ESTUDIO COMPARATIVO DE TÉCNICAS DE ENTRENAMIENTO Y CLASIFICACIÓN ENSISTEMAS DE DETECCIÓN DE INSTRUSOS (IDS), BASADOS EN ANOMALIAS DE RED.

Ing. Kevin Ibáñez

Corporación Universidad de la Costa

CUC

Maestría en Ingenierías

Barranquilla - Colombia

2017

ESTUDIO COMPARATIVO DE TÉCNICAS DE ENTRENAMIENTO Y CLASIFICACIÓN EN SISTEMAS DE DETECCIÓN DE INSTRUSOS (IDS), BASADOS EN ANOMALIAS DE RED.

Ing. Kevin Ibáñez

Tutor:

Ing. Fabio Mendoza Palechor, MSc.

Cotutor:

Ing. Paola Ariza Colpas, MSc

Corporación Universidad de la Costa

CUC

Maestría en Ingenierías

Barranquilla- Colombia

2017

Resumen

La principal motivación de esta investigación ha sido la implementación del método Draper aplicado a los sistemas de detección de intrusos en distintas técnicas de entrenamiento y clasificación con el propósito de identificar el mejor modelo de detección de intrusiones con el objetivo de mejorar las tasas de detección de ataques en sistemas de redes computacionales, utilizando un procedimiento de selección de características y distintos métodos de algoritmos de entrenamientos no supervisados, en este caso se utilizó la técnica INFO.GAIN identificando que el número de características óptimo es 15. En consecuencia, se entrenó una red neuronal que utilizan un algoritmo de aprendizaje no supervisado (GHSOM, RANDOM FOREST, REDES BAYESIANAS, NAIVE BAYES, C4.5,LOGISTIC, PART Y NBTREE), con el propósito de clasificar el tráfico bi-clase de forma automática, Como resultado se obtuvo que la mejor técnica de entrenamiento y clasificación utilizando la técnica de selección INFO.GAIN a 15 características y validación cruzada 10 pligues, fue la técnica RANDOM FOREST.

Palabras clave **:**Dataset DARPA NSL-KDD, Sistema de Detección de Intrusiones - IDS, Técnicas de entrenamiento y clasificación, Técnicas de selección de características.

Abstract

The main motivation of this investigation was the implementation of the Draper method applied to intrusion detection systems in different training and classification techniques in order to identify the best intrusion detection model with the objective of improving detection rates of attacks in computer network systems, using a procedure of selection of characteristics and different methods of algorithms of unsupervised trainings, in this case was used the technique INFO.GAIN identifying that the number of optimal characteristics is 15. Consequently, a neural network using a non-supervised learning algorithm (GHSOM, RANDOM FOREST, BAYESIAN NETWORKS, NAIVE BAYES, C4.5, LOGISTIC, PART AND NBTREE) for the purpose of classifying bi-class traffic automatically. obtained the best technique of training and classification using the selection technique In INFO.GAIN with 15 characteristics and cross validation 10 pligues, was the RANDOM FOREST technique.

Keywords: Dataset DARPA NSL-KDD, Intrusion detection system-IDS, Training techniques and classification, Feature selection.

Contenido

Lista de tablas

Lista de figuras

1.Introducción

A continuación, se presenta una breve síntesis que contextualiza al lector, con el tema objetivo de estudio, describiendo la principal motivación que propicio el desarrollo de esta investigación, en relación a los sistemas de detección de intrusiones basados en anomalías de red, culminando con una descripción general de la organización estructural de la memoria.

1.1 tópico principal y problemática abordada

Las entidades requieren obligatoriamente proteger la información que almacenan y que transita por sus redes informáticas. Teniendo en cuenta la anterior problemática han surgido distintos sistemas para detectar y proteger los datos ante él envió de malware, que podría causar la pérdida o deterioro de la información. Sin embargo, estos sistemas pueden no ser totalmente eficaces en procesos de detección de ataques de red, cuando la base de datos de malware no se actualiza con cierta frecuencia, y dado que cada vez se crean con mayor periodicidad nuevos ataques, esto puede generar muchas vulnerabilidades en el transcurso del tiempo.

Existen diferentes técnicas para prevenir y corregir acciones intrusivas maliciosas, entre ellas: Listas de Control de Acceso (ACLs), Encriptamiento de mensajes, Bloqueo de puertos, Redes Privadas Virtuales (VPNs) y Cortafuegos (firewalls). Estos últimos restringen el tráfico de servicios desconocidos, mediante el bloqueo de puertos. Si bien, son útiles para contrarrestar una gran variedad de ataques, queda aún un hueco de seguridad

desde el exterior, cuando se encapsulan los ataques en el tráfico de servicios permitidos por el dispositivo, además, tanto los firewalls como las otras técnicas mencionadas, no controlan los ataques que se generan desde el interior de la red.

Para solventar estos inconvenientes se han desarrollado Sistemas de Detección de Intrusos (IDS) que identifican tráfico malicioso en la red para un posterior proceso de bloqueo y documentación que contrarreste las acciones del atacante. Los IDS pueden detectar ataques con una metodología basada en firmas (comparando los ataques con una base de datos de firmas o rules) o con una metodología basada en anomalías (empleando un algoritmo de aprendizaje), los primeros se han implementado ampliamente en IDS comerciales y en software libre (Snort y Prelude), sin embargo, no detectan ataques nuevos; los segundos detectan ataques nuevos con cierto porcentaje de exactitud.

Para poder identificar con precisión la magnitud del problema y las posibles alternativas de solución, se deben abordar con detalle: la fundamentación referida a los Sistemas de Detección de Intrusos, las características inherentes a los dataset DARPA, las técnicas o algoritmos existentes en relación con la selección y extracción de características y técnicas de entrenamiento y clasificación de datos, tales como Redes Neuronales Artificiales (Artificial Neural Network - ANN) y Redes Bayesianas (BAYES –NET), entre otras.

Producto de las experimentaciones efectuadas, los investigadores han detectado que una variable que incide directamente en la eficiencia del algoritmo de aprendizaje, es la identificación de las características que se van a evaluar durante la fase de pre-procesamiento, debido a que la escogencia de la totalidad de características o algunas de

ellas que no sean las apropiadas, generará largos tiempos de respuesta computacional, incidiendo negativamente en la evaluación final del algoritmo de aprendizaje.

El modelo propuesto entrena una red neuronal que posteriormente, de forma automática, efectúa el proceso de clasificación de flujos de datos. Tal red neuronal es capaz de identificar el tipo de tráfico, independientemente de si se generan nuevos tipos de ataques.

Para la validación del modelo se implementaron varios escenarios de simulación, los cuales comprenden tres fases (entrenamiento, clasificación y cálculo de métricas).

Inicialmente se aplicó la técnica de análisis de correlación al dataset KDD DARPA-Train para eliminar las características de menor relevancia, a su vez se aplicó la técnica de selección de características INFO.GAIN, con el fin de identificar la cantidad optima de características, para así categorizarlas por orden de relevancia, y así efectuar posteriormente el entrenamiento de la red neuronal con las características ya anteriormente seleccionadas.

En la fase de clasificación se aplicó la técnica de validación cruzada a 10 pliegues utilizando el dataset anteriormente mencionado (KDD DARPA Train) en donde se aplican las técnicas de análisis de correlación y la técnica de selección de características INFO.GAIN, por último, se clasifican los datos, basándose en el mapa generado en el proceso de entrenamiento y en el nuevo subconjunto de datos.

En la fase final se calcularon diferentes métricas de desempeño (sensibilidad, especificidad, precisión y exactitud), esto permitió determinar la eficiencia del modelo planteado.

El tema de estudio genera un positivo impacto científico, sentando las bases de una futura implementación del modelo propuesto de detección de intrusiones en sistemas de red, en IDS comerciales, lo que posibilitará y favorecerá los procesos de detección y clasificación de tráfico normal y anómalo, de forma no supervisada, suprimiendo la necesidad de una actualización manual (por parte de un especialista humano) de la base de datos de ataques.

1.2 Mapa del documento

El presente documento de investigación está constituido por seis (6) capítulos, cada uno de los cuales brevemente se describe a continuación:

Capítulo primero:

En este capítulo se introduce al lector en el tópico principal y la problemática a abordar en el desarrollo de la investigación propuesta.

Capítulo segundo:

En este capítulo se abordan los ejes temáticos que fundamentan la investigación, en relación a: la seguridad informática, la aplicaciones de los datasets a procesos que impliquen el uso de Sistemas de Detección de Intrusos (IDS), el proceso de simulación aplicado a los IDS, las técnicas de minería de datos aplicadas al desarrollo de sistemas de detección de anomalías, las fases de pre-procesamiento, entrenamiento y clasificación de la información que se requieren implementar en un IDS y la evaluación de las métricas de desempeño para determinar la eficiencia de un IDS.

En el preprocesamiento se detalla el proceso de normalización y se especifican las técnicas de selección de características utilizadas en el modelo propuesto. En cuanto al entrenamiento y la clasificación, se destaca la importancia de técnicas de minería de datos,

aplicadas a los sistemas de detección de anomalías, específicamente la implementación de las redes neuronales como técnicas de aprendizaje automatizado.

Capítulo tercero:

En este capítulo se plasman los objetivos y la pregunta problema a desarrollar.

Capítulo cuarto:

En este capítulo se desarrolla la propuesta de IDS basado en anomalías de red, efectuando una breve descripción del modelo, se define su estructura funcional y se describen cada una de las fases del modelo propuesto.

Capítulo quinto:

En este capítulo se desarrollan los diferentes escenarios experimentales de acuerdo con las distintas técnicas de entrenamiento y clasificación seleccionadas en la investigación, lo cual permitió realizar el estudio comparativo de eficiencia de métricas y eficacia del modelo hibrido propuesto.

Capítulo sexto:

Este capítulo contiene las conclusiones a las cuales se ha llegado producto del desarrollo de la tesis de investigación, en la cual se enuncian los resultados obtenidos y se plantean los trabajos futuros que proyectan una continuidad de la línea de investigación objeto de estudio.

2. Revisión de la literatura

2.1 Seguridad informática

La primera definición generalizada de seguridad informática la plantea (Garfinkel, Spafford, & Schwartz, 2003), como la protección contra el comportamiento inesperado basándose en un conjunto de procedimientos y tecnologías orientadas a evitar intrusión.

Según (Russell & Gangemi, 1991) la Seguridad Informática es el cumplimiento de las premisas de confidencialidad, integridad y disponibilidad en un sistema informático, fundamentándose en una serie de elementos conceptuales que es necesario detallar, para una mayor comprensión de los temas de estudio.

Según los conceptos anteriores podemos inferir que la seguridad informática consiste en asegurar que los recursos del sistema de información (material informático o programas) de una organización, sean utilizados de la manera que se decidió y que el acceso a la información allí contenida, así como su modificación, sólo sea posible a las personas que se encuentren acreditadas y dentro de los límites de su autorización.

2.1.1 Premisas básicas relacionadas con la seguridad informática

La seguridad informática se fundamenta en una serie de elementos conceptuales que es necesario detallar para una mayor comprensión de los temas a abordar. Tales elementos son los mecanismos de seguridad, utilizadas en contextos organizativos. Según (Vieites, 2007) los mecanismos existentes para la protección informática, son un conjunto de recursos destinados a lograr que los activos de una organización sean confidenciales, íntegros, consistentes y disponibles a sus usuarios, autentificados por mecanismos de control de acceso y sujetos a auditoria.

Un sistema se considera seguro si cumple con las propiedades de integridad, identificación, control de acceso, no repudio, confidencialidad y disponibilidad de la información. Cada una de estas propiedades conlleva la implementación de determinados servicios y mecanismos de seguridad las cuales se describirán a continuación:

Integridad:

Este principio garantiza la autenticidad y precisión de la información sin importar el momento en que se solicita, es decir una garantía de que los datos no han sido alterados ni destruidos de modo no autorizado.

Confidencialidad:

Según (Feiertag, Kahn, & Porras, 1999), se define como "el hecho de que los datos o la información esté únicamente al alcance de las personas, entidades o mecanismos autorizados, en momentos autorizados y de una manera autorizada".

Relación entre confidencialidad e integridad:

El flujo de información se puede controlar para incrementar la seguridad mediante la aplicación de modelos como el de (Bell & LaPadula, 1973) para proveer confidencialidad y el modelo BIBA (Biba, 1977) para proveer integridad. Ambos modelos son conservadores y restringen operaciones de lectura y escritura para asegurar que no se pueda comprometer la integridad y la confidencialidad de los datos de un sistema. Por ello, un sistema completamente seguro no sería de gran utilidad ya que sería demasiado restrictivo (Kumar, 1995).

El problema de la confidencialidad se vincula comúnmente con técnicas denominadas "encriptación" y la autenticidad con técnicas denominadas "*firma digital*". Aunque la solución de ambos, en realidad, se reduce a la aplicación de procedimientos

criptográficos de encriptación y des-encriptación. Este mecanismo únicamente limita el acceso a un objeto en el sistema, pero no modela ni restringe qué es lo que un sujeto puede hacer con el objeto en el caso de que tenga acceso a su manipulación (Denning, 1992).

Disponibilidad:

Según(Cestero, 2013), la disponibilidad es el "grado en el que los datos están en el lugar, momento y forma en que es requerido por el usuario autorizado". Situación que se produce cuando se puede acceder a un sistema de información en un periodo de tiempo considerado aceptable. La disponibilidad está asociada a la fiabilidad técnica de los componentes del sistema de información.

Autenticación (identificación):

El sistema debe ser capaz de verificar que un usuario identificado, que accede a un sistema o que genera una determinada información, es quien dice ser. Solo cuando un usuario o entidad ha sido autenticado, podrá tener autorización de acceso. Se puede exigir autenticación en la entidad de origen de la información, en la de destino o en ambas. (Piattini & Peso, 2001).

Irrenunciabilidad:

Proporcionará al sistema una serie de evidencias irrefutables de la autoría de un hecho. El no repudio consiste en no poder negar haber emitido una información que efectivamente se ha emitido y en no poder negar su recepción cuando ha sido recibida.

2.1.2 Taxonomía de los ataques informáticos e intrusiones.

Una taxonomía permite tener un conocimiento previo que se aplicará a nuevos ataques, así como proporciona una forma estructurada para el estudio de estos ataques. Se

pueden encontrar multitud de trabajos referentes a la categorización y clasificación de ataques informáticos e intrusiones (Howard, 1997) .

Según lo propuesto por el modelo de (Kendall, 1999), y los ataques informáticos contenidos en el DATASET DARPA (Singh, Singh, & BUIT, 2014) tratado más afondo en la sección **2.5** se clasifican en cuatro (4) categorías:

Denegación de servicio:

Denominado también por sus siglas en ingles *Denial of Service* (DoS), son un conjunto de ataques que conllevan a detener el funcionamiento de una red, máquina , proceso o servicios a usuarios autorizados debido a la sobrecarga de los recursos computacionales de la víctima (Marchette, 2001).

Durante este tipo de ataques se saturan los puertos de comunicación con excesivo flujo de datos, de manera tal que la sobrecarga del sistema haga imposible la correcta prestación del servicio, negando en la mayoría de los casos diferentes peticiones efectuadas por usuarios que las solicitan.

Remote to Local (R2L):

Según (Sabhnani & Serpen, 2003) se origina cuando un atacante informático que no posee acceso a alguna máquina, logra acceder a dicho equipo ya sea como usuario común o root, utilizando algún método de intrusión o utilizando programas .En la mayoría de los ataques R2L se ingresa a través del servicio de internet causando en la mayoría de los casos desbordamiento de buffer, debido a la mala implementación de políticas de seguridad en los sistemas de red.

User to Root (U2R):

Se genera cuando un atacante, que tiene una cuenta en un sistema informático, adquiere privilegios superiores a los inicialmente establecidos, sin autorización del administrador de TI, ejecutando alguna técnica de intrusión que se basa en una determinada vulnerabilidad del sistema informático. Por ejemplo, cuando se genera un acceso intrusivo al sistema operativo y posteriormente se instala algún tipo de software para acceder a mayores privilegios (Chou & Yen, 2007) .

Esto se genera debido a las vulnerabilidades de los sistemas operativos o de las aplicaciones no configuradas con altos estándares de seguridad, a la falta de concientización de los usuarios o en su defecto la instalación de programas que actúan como espías que posibilitan el acceso intrusivo.

Probing*:*

Según (Brugger & Chow, 2007), son un conjunto de ataques que se caracterizan por sondear la red de la víctima para recopilar información importante de los host que la incluye sin ser detectada, previéndole al atacante información necesaria para así tener una lista de vulnerabilidades potenciales y llevar a cabo un ataque informático a los servicios como a las maquinas que lo ejecutan.

Un atacante con un mapa de las máquinas y servicios disponibles en una red puede utilizar esta información para encontrar todos los puntos débiles de esta última. Algunas de estas herramientas de análisis "satan, saint, mscan” permiten que incluso un hacker principiante, pueda revisar rápidamente cientos o miles de máquinas en una red.

2.2 Sistemas de detección de intrusos

Según (Dokas et al., 2002) los IDS son una medida de seguridad, que ayuda a identificar un conjunto de acciones malintencionadas que comprometen la integridad,

confidencialidad, y la disponibilidad de los recursos de informáticos. Esta tesis se centra en la detección de acceso no autorizado, razón por la cual se aborda con detalle la funcionalidad de los IDS.

La función principal de los IDS es proteger la información de las organizaciones frente a cualquier amenaza. Estas se han visto reflejadas en una creciente ola de ataques informáticos debido al incremento en la demanda del uso de las redes, el internet y la dependencia de los sistemas de información, es por ello que (Bace & Mell, 2001)plantea seis(6) razones de peso para adquirir un IDS:

1. Evitar problemas de conducta mediante el aumento de la percepción del riesgo descubierto y castigar a los que atacarían o de otra manera abusarían del sistema.
2. Detectar ataques y otras violaciones de seguridad que no son detectadas por otras medidas de seguridad.
3. Descubrir y tratar con los preámbulos necesarios los ataques o intentos de ataques.
4. Documentar las amenazas existentes en una organización.
5. Actuar como el control de calidad para el diseño y administración de seguridad especialmente en empresas grandes y complejas.
6. Proporcionar información útil sobre las intrusiones que tienen lugar, lo que permite un mejor diagnóstico, recuperación y corrección de los factores causales.

2.2.1 Clasificación de los sistemas de detección de intrusos

Según (Bace & Mell, 2001) y (Dokas et al., 2002) los IDS se clasifican de acuerdo a la estrategia o tipo de análisis, fuente de información, arquitectura o estructura, respuesta o comportamiento o tipo de predicción. La Figura **No 2.1** detalla el esquema de clasificación de los IDS.

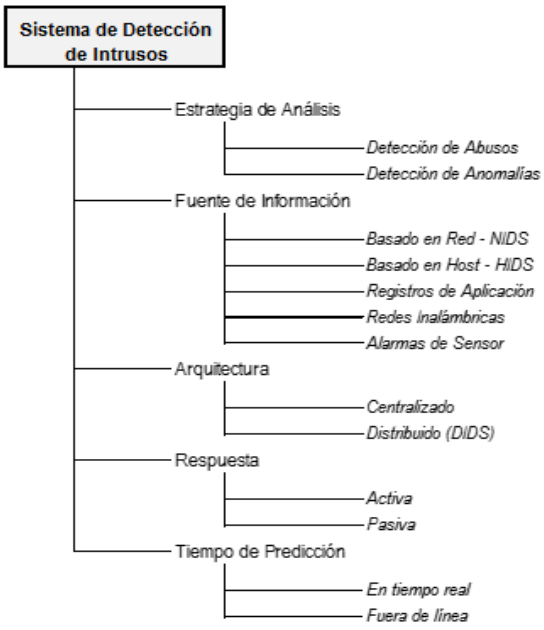


*Figura No. 2. 1.*Clasificación de IDS

Fuente: (De la hoz, 2014)

Las metodologías de detección de intrusos se dividen en 3 categorías detección basada en abuso, detección basada en anomalías y análisis de protocolo de estado las cuales han sido nombradas por diferentes autores(Bace & Mell, 2001),(Dokas et al., 2002),(De la hoz, 2012):

- **Detección Basada En Abusos**:

  También se conoce como detección basada en el conocimiento, detección de uso incorrecto o detección basada en firmas. Este tipo de detección analiza la actividad del sistema en busca de uno o varios eventos que coincidan con un patrón predefinido y que describan a un determinado ataque.

Tales eventos son conocidos como firmas, las cuales son patrones o cadenas, que corresponde a un ataque o amenaza conocida. En el caso de que se presente un ataque que coincida con una o varias firmas, se genera una alarma(Bace & Mell, 2001).

- **Detección Basada En Anomalías (AD)**:

  Una anomalía es una desviación a un comportamiento conocido, y los perfiles representan los comportamientos normales o esperados, derivados del seguimiento de las actividades regulares, conexiones de red, hosts o usuarios, durante un período de tiempo. Los perfiles se dividen en dos (2) tipos: *Estáticos* o *Dinámicos*. Son registros del sistema que se alimentan de una serie de atributos o características, como son: el número de intentos fallidos de inicio de sesión, el uso del procesador, el recuento de los correos electrónicos enviados, entre otros.

  La detección basada en anomalías realiza una comparación entre eventos y perfiles normales de eventos; posteriormente, son observados para reconocer ataques significativos. AD en algunos artículos, es conocida como detección basada en comportamiento. Algunos ejemplos de AD se presentan en (Liao, Lin, Lin, & Tung, 2013). Este tipo de detección tiene una medida y técnica comúnmente utilizada para detectar anomalías, estas son: detección de umbral y uso de medidas estadísticas, se puede ampliar información de cada una de estas en los siguientes artículos (De la hoz, 2012), (De la hoz, 2014), (Bace & Mell, 2001) y (Guo, Zhou, Ping, Luo, & Lai, 2013).

- **Análisis De Protocolo Con Estado (SPA)**:

  Indica que un IDS podría conocer y rastrear los estados de protocolo (por ejemplo, las solicitudes de emparejamiento de las respuestas). SPA se parece a AD, pero son esencialmente diferentes. AD adopta la red precargando perfiles específicos

en host, mientras SPA depende de perfiles genéricos desarrollados sobre protocolos específicos. En general, los modelos de protocolos de red en SPA se basan, originalmente, en las normas de organizaciones internacionales estándares, por ejemplo, el IETF.

La mayoría de los IDS-híbridos utilizan varios métodos para que la detección sea amplia y precisa. Por ejemplo, detección basada en firmas y detección basada en anomalías son métodos complementarios, ya que el primero se refiere a determinados ataques o amenazas y el último se centra en ataques desconocidos, la Tabla **No. 2.5** muestra las ventajas y desventajas de cada uno de ellos.

Tabla No.2.1.*Ventajas y desventajas según el tipo de análisis de IDS*

DETECCIÓN BASADA EN ABUSOS	**DETECCIÓN BASADAS EN ANOMALÍAS (AD)**	**ANÁLISIS DE PROTOCOLOS (SPA)**
Pros - Sencillo y eficaz método para detectar ataques conocidos. - Análisis contextual detalle.	**Pros** - Efectiva para detectar nuevas vulnerabilidades e imprevistas. - menos dependiente del sistema operativo. - Facilitar la detección del abuso de privilegio.	**Pros** - Conocer y localizar los estados de protocolo. - Distinguir secuencia inesperada de comandos.
Contras - Ineficaces para detectar ataques desconocidos, ataques de evasión, y variantes de ataques conocidos.	**Contras** - Perfiles débiles precisión debido a eventos observados constantemente cambian. - No disponible durante la reconstrucción de los	**Contras** - Recursos consumo excesivo de rastreo de estado de protocolo y el examen.

• Poca comprensión de los estados y los protocolos. • Difícil mantener firmas / patrones hasta la fecha. • Requiere mucho tiempo para mantener los conocimientos.	perfiles de comportamiento. • Difícil de activar alertas en tiempo correcto.	• No es posible inspeccionar los ataques a la vista como los Comportamientos de protocolo benignos. • Podría incompatibilidad de los sistemas operativos o los puntos de acceso dedicado.

Fuente:(Liao et al., 2013)

Híbridos:

Como se aprecia en la Tabla **No. 2.1**, los IDS basados en detección de abusos como los basados en anomalías, presentan pros y contras. Los primeros son más confiables debido a que generan un rendimiento más óptimo al momento de detectar ataques conocidos, ya que están basados en una firma definida; infortunadamente, no tienen la capacidad de reconocer ataques que no estén incluidos en su base de datos. Los segundos, tienen la ventaja de detectar un ataque desconocido pero su rendimiento es inferior al primero.

Es por ello, que el modelo indicado para mitigar las desventajas y aprovechar las ventajas de cada uno, sería la combinación de un modelo hibrido con capacidad de detección de ataques previamente predefinidos e interceptar ataques para el sistema que por tanto no estén incluidos en sus firmas.

Un ejemplo de IDS hibrido es el *Minesora Intrusion Detection System- MINDS*, definido en (Ert, Eilertson, Tan, Kumar, & Srivastava, n.d.). Este IDS fue desarrollado en la Universidad de Minnesota, el cual integra un sistema de detección basado en abusos con uno basado en anomalías, utilizando un algoritmo denominado *Shared Nearest Neighbour– SNN*.

Cuando el sistema detecta las anomalías realiza un análisis que asocia patrones, de esta manera recoge características las cuales determinan si se trata de un ataque. En caso de detectarlas, las añade a la base de datos como una nueva firma.

Tipos de tecnología:

Actualmente, existen una gran variedad de tecnologías enfocadas a los IDS, éstas han sido clasificadas en cuatro (4) categorías dependiendo del tipo de fuente que usa el IDS para capturar la información. Algunas tecnologías analizan paquetes de red, los cuales pueden ser capturados de entornos como una red LAN o una troncal, con el objetivo de encontrar un posible atacante o ataque.

Otras, su fuente principal son reportes generados por aplicaciones (logs)para ser analizadas y encontrar señales de intrusión (Stavroulakis & Stamp, 2010)(Bace & Mell, 2001)(Liao et al., 2013).La Tabla **No.2.2**describe la funcionalidad, ventajas y desventajas de las diferentes tecnologías de IDS.

Tabla No.2.2.*Ventajas y desventajas Según tipo de Tecnología de IDS.*

	IDS Basados en host (HIDS)	**IDS Basados en red (NIDS)**	**Wireles IDS (WIDS)**	**IDS Mixtos (MIDS)**
FUNCIÓN	• Radica en evaluar la información que generan los usuarios, el sistema operativo y las aplicaciones de una computadora (host), conectada a una red informática; con el objetivo de identificar amenazas e intrusiones ejecutadas a nivel del host local.	• Detectar ataques capturando y analizando los paquetes de Red	• La función del WIDS es similar a la del NIDS a diferencia que su entorno es en redes inalámbricas	• Mezcla las tecnologías anteriores ofreciendo una detecciónmás completa y exacta
VENTAJAS	• Su funcionamiento no se ve afectado por redes conmutadas. • Debido a la capacidad de supervisar los eventos que ocurren localmente en un host puede detectar ataques que no son detectados por un NIDS.	• Bien colocados pueden administrar una Red grande. • Son muy seguros contra el ataque e incluso ser invisibles para el atacante.	• Al momento de desplegarlos en un entorno operacional no suelen tener mayor impacto en el buen funcionamiento de la red.	• Combinación de todas las anteriores

DESVENTAJAS	▪ Altos consumos de recursos de HOST. ▪ Retrasos en generación de alertas	▪ No analiza información cifrada. ▪ No monitorea protocolos inalámbricos	▪ No analiza información cifrada. ▪ Los sensores son susceptibles a ataques de interferencia física	▪ Limitación de acceso debido a su alto costo

Fuente: (Stavroulakis & Stamp, 2010)(Bace & Mell, 2001)(Liao et al., 2013)

2.2.2. Arquitectura de un IDS

En esta sección se define la arquitectura de un IDS ,la cual puede ser centralizada o distribuida según(Liao et al., 2013), y se presentan según representación esquemática.

Arquitectura de IDS centralizado

Los sistemas centralizados capturan la información desde diferentes sensores, que la retrasmiten a un IDS Central o administrador para un posterior análisis de ésta. En la Figura **No.2.2** se observa la estructura de un IDS centralizado.

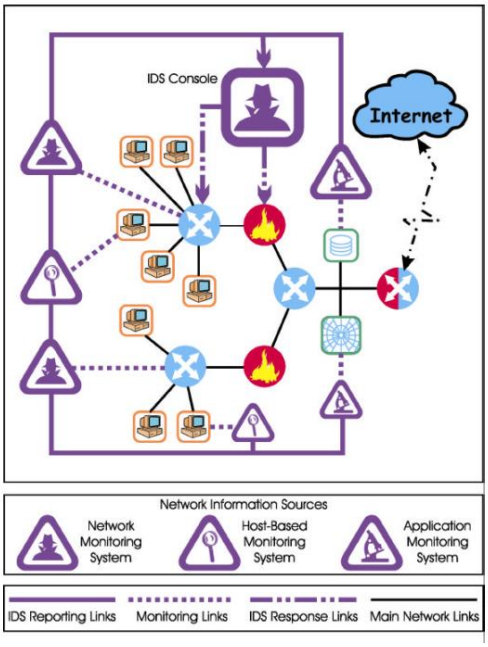


Figura No. 2. 2. Arquitectura de un IDS Centralizado

Fuente: (Bace & Mell, 2001)

ARQUITECTURA DE IDS DISTRIBUIDO

Los sistemas distribuidos se basan en la topología cliente servidor, en donde múltiples clientes son ubicados estratégicamente en una red, con el fin de analizar todo el tráfico que circula través de ella. Este tipo de arquitectura fue desarrollada para subsanar las debilidades de los IDS centralizados. La Figura **No.2.3** tomada de (Bace & Mell, 2001)muestra la estructura de un IDS distribuido.

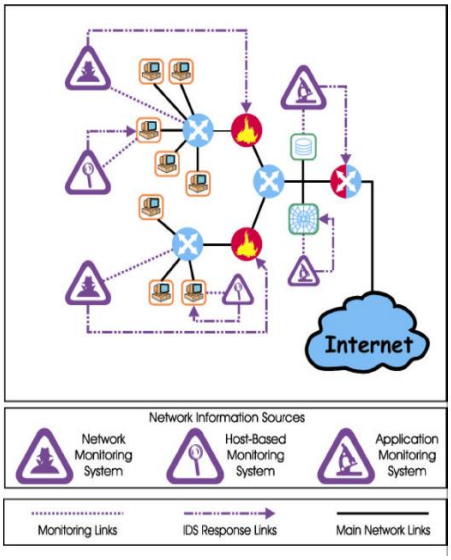


Figura No. 2. 3. Arquitectura de un IDS Distribuido(Bace & Mell, 2001)

Fuente: (Bace & Mell, 2001)

2.2.3. IDS según el tipo de respuesta

Cuando unIDS detecta una anomalía y posteriormente la analiza para identificar un ataque, genera una respuesta que puede ser **pasiva** o **activa**. Las respuestas generadas se almacenan en una serie de archivos tipo log, los cuales se encuentran ubicados en una determinada ruta de almacenamiento y contienen las conclusiones del análisis.

IDS de respuesta pasiva

Este tipo de IDS no puede ejecutar acciones de forma autónoma y con propósitos correctivos o con la intensión de mitigar o finalizar la ejecución del ataque. Los IDS de respuesta pasiva, una vez detectado el ataque, notifican al administrador de la seguridad de la red informática, por medio de una alarma y documentan el ataque.

IDS de respuesta activa

Estos sistemas ejecutan acciones automatizadas cuando detectan ciertos tipos de ataques. Las características de los IDS de respuesta activa, según (Bace & Mell, 2001) y (De la hoz, 2012) son: recoger información adicional, generar cambios de ambiente y tomar acciones sobre el atacante. Cada una de estas características, se amplía a continuación:

- **Recoger información adicional**

  Una vez el IDS detecta un ataque aumenta su nivel de sensibilidad, analizando con mayor detalle las fuentes de información, es decir, los datos que provienen de los nodos sensores, con el fin de obtener más información, en relación al posible ataque.Según (Bace & Mell, 2001) la recopilación de información adicional es útil por varias razones: (1) ayudar al sistema en el diagnóstico de un ataque, es decir, si éste fue o no generado; (2) reunir información que puede ser utilizada para apoyar la investigación, como evidencia legal para un posterior proceso penal o civil, con miras a la aprehensión del atacante.

- **Cambio de ambiente**

  Según (Bace & Mell, 2001) este tipo de IDS, de respuesta activa, es capaz de detener un ataque en curso y posteriormente bloquear el acceso al atacante. Para cumplir con su objetivo, estos sistemas toman las siguientes medidas: (1) inyectar paquetes TCP a la conexión del atacante, para finalizarla y detener el ataque; (2) reconfiguración de routers y firewalls de red, mediante el bloqueo de puertos, protocolos o servicios utilizados por el atacante.

- **Tomar acciones sobre el atacante**

Consiste en lanzar ataques en contra del intruso o intentar activamente obtener información sobre el dispositivo utilizado por el atacante o el sitio donde éste se encuentra.

2.2.3. Fundamentos relativos a la evaluación de los IDS

En la actualidad no existe un IDS 100% efectivo que clasifique a la perfección el tráfico normal del malicioso, debido a que hay una gran variedad de ataques, que con el pasar del tiempo se incrementan, siendo cada vez más novedosos y desconocidos para los IDS. A esto se suman las malas prácticas en **tecnologías de la información.** Lo que puede generar que un IDS tome decisiones incorrectas, en procesos de clasificación del tráfico de red. Por ejemplo: identificar ataques como tráfico normal.

Según (De la hoz, 2012), para evaluar el desempeño de un IDS se han identificado cuatro métricas asociadas a la naturaleza del evento (tráfico inofensivo o ataque) y al estado de la detección (normal o anómala), tal como se aprecia en la Figura **No. 2.4**.

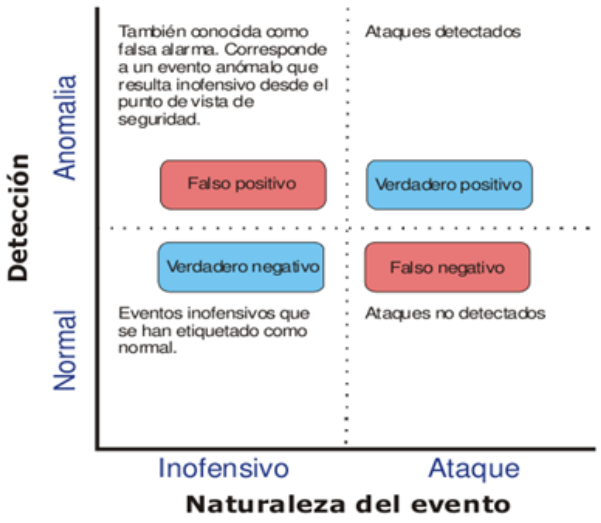


Figura No. 2. 4. Matriz de Confusión

Fuente: (De la hoz, 2014)

Un IDS es más eficiente cuando durante el proceso de clasificación del tráfico de datos presenta mayores tasas de aciertos (es decir, el porcentaje de verdaderos negativos y verdaderos positivos tiende a 100%) y consecuentemente, presenta bajas tasas de fallos (es decir, el porcentaje de falsos positivos y falsos negativos tiende a 0%).

A partir de lo anterior, se concluye que un IDS perfecto es aquel que detecte todo el tráfico de forma correcta, no generando ninguna falsa alarma. Según(De la hoz, 2014) lo anterior se puede inferir que:

- **Verdaderos Positivos (VP):**

  Ataque correctamente detectado como anomalía.

- **Falsos Positivos (FP):**

  Tráfico inofensivo detectado de forma incorrectamente como anomalía.

- **Verdaderos Negativos (VN):**

  Tráfico inofensivo correctamente identificado como tráfico normal.

- **Falso Negativo (FN):**

  Ataque identificado incorrectamente como tráfico normal.

2.2.4. Métricas de desempeño

En esta investigación se usan métricas de desempeño estadísticas, para medir el comportamiento del IDS en relación al proceso de clasificación, en coherencia con lo planteado por (Andersen, Glasdam, & Larsen, 2016),(Eid, Hassanien, Kim, & Banerjee, 2013) y (Panda, Abraham, & Patra, 2010);tales métricas se definen a continuación.

- **Sensibilidad**:

(De la Hoz & De la Hoz, 2012) define sensibilidad como la capacidad que tiene un IDS para identificar resultados "*verdaderos positivos*":

$$Sensibilidad = \frac{VP}{VP + FN} \qquad (2.1)$$

- **Especificidad**:

  (De la Hoz & De la Hoz, 2012) define especificidad como la capacidad que tiene un IDS de medir la proporción de "*verdaderos negativos*" que se han identificado correctamente:

$$Especificidad = \frac{VN}{VN + FP} \qquad (2.2)$$

- **Exactitud**:

  (Andersen et al., 2016) define exactitud como el grado de cercanía de las mediciones de una cantidad (X) al valor de la magnitud real (Y); Es decir la proporción de resultados verdaderos (tanto verdaderos positivos como verdaderos negativos). Una exactitud del 100% significa que los valores medidos son exactamente los mismos que los valores dados:

$$Exactitud = \frac{VP + VN}{VP + FP + FN + VN} \qquad (2.3)$$

- **Precisión**:

  Define la proporción de verdaderos positivos contra todos los resultados positivos:

$$Precisión = \frac{VP}{VP + FP} \qquad (2.4)$$

2.3 Proceso de simulación aplicado a los IDS

Para la efectividad en un proceso de detección de tráfico malicioso en una red computacional implementando un sistema de IDS que utilice técnicas de selección de características, algoritmos de aprendizaje y medición de calidad de métricas, es idónea la evaluación mediante simulación por software en laboratorio.

Por ende, requiere la ejecución de varias fases, como se muestra en la Figura**No.2.5**, las cuales son: Elección de la colección de datos (dataset), preprocesamiento (parseo y normalización), selección de características, entrenamiento (training), clasificación y evaluación de métricas.

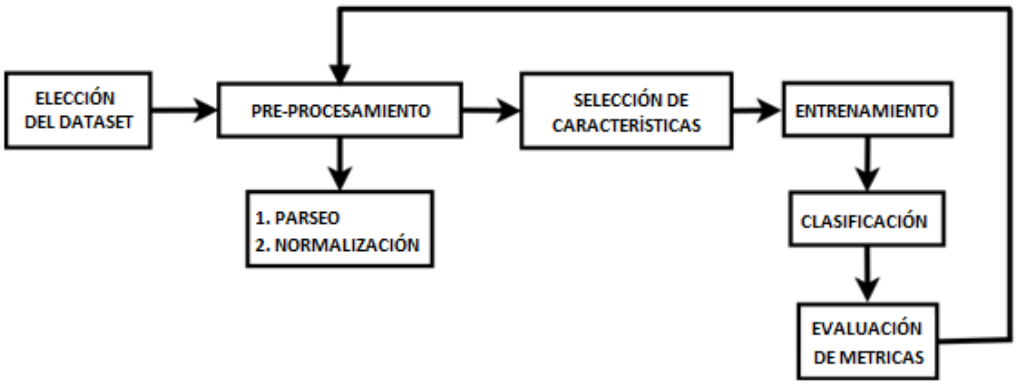


*Figura No. 2. 5.*Fases de simulación

Fuente: Construcción propia.

2.3.1. Fase de elección de la colección de datos (dataset)

En esta fase inicial se debe seleccionar la colección de datos que se va a utilizar para las siguientes fases. Existen distintas fuentes de datos utilizadas en sistemas de detección de intrusos, por ejemplo: PREDICT, DARPA KDD-NSL, CAIDA, CRAWDAD, DRDC, NIST SAMATE y Virtual dataset repository.

2.3.2. Fase de pre-procesamiento

El pre-procesamiento es una fase previa a la selección, reducción o extracción de características en el proceso de simulación en el cual se emplea una técnica de clasificación basada en redes neuronales artificiales para su aprendizaje no supervisado.

Esta fase permite homogenizar la presentación de los datos provenientes del dataset e integrar esos datos contenidos en un formato diferente a la herramienta de simulación que se va a utilizar para efectos de procesar los datos. El pre procesamiento implica la ejecución de dos (2) sub fases llamadas Parseo o Parsing y normalización:

- El parsing se refiere, básicamente a presentar los datos que proceden del dataset RAW (en bruto u original), el cual está formateado en .txt, a un formato que sea fácil de procesar por la herramienta en la cual se implementa la simulación, que en este caso es Matlab™. Ello requerirá recorrer iterativamente cada registro de la colección de datos del archivo .txt, que tiene una estructura del tipo CVS (con separación de los datos por comas), extrayendo registro a registro de datos no estructurados a un formato de datos estructurados, de tal forma que estos se representen como una matriz de datos donde cada fila corresponde a un patrón o una

conexión de red y cada columna corresponderá a las características o atributos que permiten identificar la conexión de datos.

Una vez ejecutado el parseo, se procede a normalizar los datos, el proceso de normalización implica presentarlos en el mismo formato, dado que éstos

- inicialmente toman valores disimiles o heterogéneos. Algunos atributos toman valores simbólicos, otros toman valores discretos numéricos enteros y otros valores continuos numéricos representados en formato de coma flotante. Algunos valores son excesivamente grandes, valores numéricos representados en millones o millones de millones, y otros en valores decimales muy pequeños, de uno o dos (2) símbolos decimales. El propósito de la normalización es, por tanto, representar todos los atributos lo más homogéneo posible, para que luego de ser procesados en el proceso de entrenamiento, los datos tengan la misma representatividad en el modelo que va a efectuar el proceso de clasificación de la información.

La normalización de datos no permite que ninguna característica contribuya más que otra a la medida de la distancia. En (Vesanto, Himberg, Alhoniemi, & Parhankangas, 2000) se presentan seis implementaciones de métodos de normalización, de la SOM toolbox de Matlab™: var, range, log, logistic, histD e histC.

- **Var**, normaliza la varianza de la variable a la unidad y su media a cero. Esto es una transformación lineal simple, como se indica en la Ecuación **No. 2.7**.

$$\hat{x} = \frac{x - \bar{x}}{\sigma} \qquad (\mathbf{2.7})$$

Donde $\vec{x}$ y σ son, respectivamente, la media y la desviación estándar de la variable x. Esto es equivalente a expresar la variable x como la distancia entre el número de desviaciones estándar y su media.

- ***Range,*** escala los valores de la variable entre [0, 1] con una transformación lineal simple, como se indica en la Ecuación **No. 2.8**. Los parámetros de la transformación son: los valores mínimos y el rango $\max(x) - \min(x)$ de la variable. Si la transformación es aplicada a nuevos datos con valores por fuera del rango mínimo y máximo, los valores de la transformación estarán también por fuera del rango [0, 1].

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad (\mathbf{2.8})$$

- ***Log,*** es una transformación logarítmica útil si los valores de la variable se distribuyen de forma exponencial con valores demasiado pequeños y un número menor de valores grandes. Esta transformación es una buena forma de conseguir mayor resolución en la parte baja del vector de componentes.

  Lo que en realidad se hace es una transformación no lineal, ver Ecuación **No 2.9**, donde ln es el logaritmo natural, generando valores no-negativos. Se debe tener especial cuidado cuando la transformación es aplicada a un nuevo conjunto de datos, con valores por debajo de min(x)-1, dado que los resultados serán números

complejos. Por lo tanto, si se conocen los valores previamente, es conveniente asignarles manualmente el valor mínimo min(x).

$$x' = \ln(x - \min(x) + 1) \tag{2.9}$$

- ***Logistic***, también denominada normalización ***softmax***. Este método de normalización se asegura de que todos los valores, desde menos infinito hasta más infinito se encuentren dentro del rango [0, 1]. La transformación es más o menos lineal en la mitad del rango (alrededor del valor medio), y tiene una linealidad suavizada en ambos extremos, lo cual asegura que todos los valores estén dentro del rango.

  El dato primero se escala como una normalización de varianza, ver Ecuación **No. 2.7**, luego se aplica la función ***logistic***, ver Ecuación **No. 2.10**. Los parámetros de la transformación son el valor medio $\bar{x}$ y la desviación estándar σ de los valores originales, tal como en el método de normalización ***var***.

$$x' = \frac{1}{(1 + e^{-\hat{x}})} \tag{2.10}$$

- ***histD***, es una ecualización de histograma discreta. Que ordena los valores y luego sustituye cada uno por su número ordinal. Por último, escala linealmente los valores de modo que estén entre [0,1]. Útil tanto para variables discretas como continuas, sin embargo, como los parámetros de transformación son todos valores únicos del conjunto de datos de inicialización, esto puede requerir el uso de una considerable cantidad de memoria. Si la variable puede tomar unos pocos valores (20, por

ejemplo) puede que sea mejor utilizar el método ***histC***. El método ***histD*** no es exactamente reversible, si es aplicado a valores que no hagan parte del conjunto de valores originales.

- ***histC*** es una ecualización de histograma continua. Realmente, es una transformación lineal parcial, la cual trata de ser como una ecualización de histograma. El rango de valores se divide en una serie de contenedores de tal forma que el número de valores en cada contenedor es (casi) el mismo. Los valores son transformados linealmente en cada contenedor. Por ejemplo, valores en el contenedor número 3 son escalados entre [3,4]. Finalmente, todos los valores son linealmente escalados entre [0,1].

  El número de contenedores es el redondeo de la raíz cuadrada del número de valores únicos en el conjunto de inicialización. La ecualización de histograma resultante no es tan buena como la que hace ***histD***, pero el beneficio es que es exactamente reversible, incluso fuera del rango de valores originales.

2.3.3. Fase de selección de características

Documentada en (Hota & Shrivas, 2014), se define como el proceso de optimización que trata de encontrar el mejor subconjunto de características de un conjunto fijo de ellas.

Su objetivo es reducir el tamaño de los datos de entrada para facilitar el procesamiento y análisis, descartando datos que no contribuyen en mayor medida al

posterior proceso de clasificación. Lo cual genera un ahorro de tiempo en el procesamiento de los datos, sin desestimar la generación de resultados óptimos.

2.3.4. Fase de entrenamiento

En esta fase se procede a entrenar la red neuronal desde la implementación de algún algoritmo de aprendizaje como los son, *Mapas Auto-organizativos de Jerarquía Creciente – GHSOM, Logistic, Random Forest, Clasificadores Bayesianos, entre otros, tomando como insumo todos los registros del dataset* KDD-Train al 100%, proveniente de la aplicación de técnicas de selección de características.

2.3.5. Fase de clasificación

Una vez entrenada la red neuronal, se procede con la fase de clasificación, la cual se realiza de forma autónoma basándose en la implementación de un algoritmo de clasificación el cual determina el tráfico bi-clase (normal y anómalo), presentando la información de forma resumida y basada en fundamentos estadísticos.

Una vez culminado el proceso anterior, se procede a efectuar la prueba de aprendizaje, la cual se realiza usualmente con el dataset KDD-Test cargado al 100% de atributos.

2.3.6. Fase de evaluación de métricas

En esta última fase, se realiza una evaluación de la calidad del modelo, bajo la implementación de un algoritmo que calcula cada una de las métricas utilizadas para el análisis de trafico de red (sensibilidad, especificidad, exactitud y precisión) basándose en los resultados obtenidos en la etapa de clasificación, con el propósito de conocer las características principales del modelo planteado.

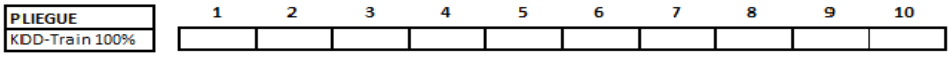
2.4 Técnica de validación cruzada aplicada en la fase de clasificación

Con el fin de demostrar que el sistema no se sobre ajusta y por lo tanto, tiene un buen rendimiento de la generalización, los procesos de selección de características y entrenamiento han sido evaluados mediante la validación cruzada k-pliegues (k = 10). Ya que k = 10, contiene particiones del 90% de las muestras que fueron seleccionadas aleatoriamente para ajustar el modelo; el resto de las muestras (10%) se utilizó para la prueba.

Como se muestra en Figura **No 2.6**, estos subconjuntos son diferentes y no comparten ninguna muestra. Este proceso se repitió para los 10 pliegues, asegurándonos que los datos de prueba nunca se hubiesen utilizado en la selección de características o en el entrenamiento del clasificador. Por lo tanto, los resultados proporcionados por los subconjuntos de características seleccionadas y la precisión de la clasificación se calculan como la media de las 10 evaluaciones a través de los 10 pliegues.

El objetivo principal de la validación cruzada es estimar el error de generalización, asegurando que resultados similares se obtuvieron en nuevos datos (es decir, bajo error de generalización). Este método calcula el error de predicción y evita la doble inmersión. Además, vale la pena señalar que, debido al elevado número de muestras de los dataset disponibles, ambos procesos de entrenamiento y de prueba se tratan usando un alto número de muestras. Esto proporciona una estimación de la varianza del error de generalización menor.

Figura No. 2. 6. Esquema de particiones para validación cruzada.



PRUEBAS				
No	**Pliegues Train**	**Pliegues Test**	**Resultados**	**Grafico**
1	2-10	1	R1	
2	1,3-10	2	R2	
3	1-2,4-10	3	R3	
4	1-3,5-10	4	R4	
5	1-4,6-10	5	R5	
6	1-5,7-10	6	R6	
7	1-6,8-10	7	R7	
8	1-7,9-10	8	R8	
9	1-8,10	9	R9	
10	1-9	10	R10	
RESULTADO FINAL			Promedio (Ri)	

Fuente: Construcción propia

2.5 Datasets en sistemas de detección de intrusos

Los investigadores siguen valorando diferentes metodologías y técnicas con el objeto de desarrollar una solución IDS cada vez mejor, para ello requieren un ambiente que permita simular el tráfico de red de la forma más real posible. Razón por la cual el *Massachusetts Institute of Technology - MIT* y *Defense Advanced Research Projects Agency - DARPA* han simulado tal escenario, alimentando colecciones de datos con el propósito de dotar a los investigadores de una base de datos de tráfico de red, que sirva de insumo para futuras investigaciones.

El dataset se utiliza para la evaluación de la eficiencia de los sistemas de detección de intrusos en redes informáticas. Los criterios medibles son la probabilidad de detección y la probabilidad de falsas alarmas del respectivo sistema analizado.

Tabla No. 2. 3. Recopilatorio de distintos datasets orientados a IDS

DATASET	***PATROCINADORES***
Dataset DARPA (Revathi & Malathi, 2013)	IST-LLMIT (Grupo de Tecnologías de Sistemas de Información- Laboratorio del Instituto de Tecnologías de Massachusetts). DARPA ITO (Agenda de Proyectos de investigación Avanzada de Defensa-Oficina de Tecnología de la información). AFRL/SNHS (Laboratorio de Investigación de las Fuerzas Aéreas).
Datasets USC/ISI ANT Programa PREDICT (Elsayed, Asadi, Wang, Lin, & Metzler, 2010)	ANT (Grupo de Investigación de Análisis de Tráfico de Red). ISI (Instituto de Ciencia de la Información). USC (Universidad del Sur de California). Departamento de Ciencias Computacionales de la Universidad estatal de Colorado. Departamento de Ingeniería Eléctrica de USC. Servicios de Tecnologías de la Información USC.
Datasets CAIDA (Hyun, Huffaker, Andersen, & Aben, 2011)	• **Patrocinadores**: ARIN (American Registry for Internet Member), CISCO, Endance Measurement Systems, U.S Departament of Homeland Security, NSF (National Science Fundation). • **Miembros:**

	▪ Digital Envoy, Intel, NIT (Nippon Telegraph and Telephone Corporation), Ripe NCC, University of California San Diego
Datasets CRAWDAD (Scott, Gass, Crowcroft, & Hui, 2006)	▪ ACM SIGMOBILE ▪ Intel Corporation ▪ Fundación Nacional de Ciencias
Dataset DRDC Defense Research and Development Canada (Sévigny, DiFilippo, & Laneve, 2010)	▪ Sección de operaciones de información de red (NIO) de la DRDC Ottawa, Canada. ▪ Red de Establecimiento para la investigación y Defensa (DREnet).
NIST SAMATE Reference Dataset Project NIST (Agosta & Barenghi, 2012)	Departamento de Estado EEUU
Virtual Dataset Repository (SINGH, 2013)	▪ MERIT NETWORK INC. Programa PREDICT (Protected Repository for the Defense of Infraestructure again Cyber threats)

Aunque existe una amplia variedad de datasets los investigadores comúnmente se han decantado por el uso de DARPA NSL-KDD en procesos de simulación de detección de intrusos tal como lo muestra la Tabla **No. 2.3**, por las ventajas que ofrecen con respecto a otros dataset de su misma familia y de otras fuentes.

En(Olson & Delen, 2008) y (ENGEN, 2010) se hace un análisis en profundidad de las discrepancias encontradas en KDD cup'99, a partir de lo cual se aprecian las mejoras obtenidas en relación a la eliminación de datos redundantes e inconsistentes en el NSL-KDD (última versión del dataset DARPA), esta información complementa el resumen de los conjuntos de datos más populares en el dominio de detección de intrusos que ha sido mostrada en (Wu & Banzhaf, 2010) y (Zargari & Voorhis, 2012).

En esta investigación se ha decidido seleccionar el dataset NSL-KDD como insumo para las posteriores fases del proceso de simulación de detección de intrusiones. Dado que El Grupo de *Tecnología de Sistemas de Información-IST,* del *Laboratorio Lincoln del Instituto Tecnológico de Massachusetts LL-MIT* ha demostrado considerables mejoras del dataset NSL-KDD respecto a sus antecesores, y la comunidad investigadora mundial (en este ámbito de conocimiento) lo ha apropiado e implementando en sus investigaciones.

2.5.1. Evolución del dataset DARPA

El descubrimiento de conocimiento en bases de datos KDD acrónimo en inglés *Knowledge Discovery in Databases-KD* , nos permite identificar eficaz y coherentemente, patrones potencialmente útiles y previamente desconocidos en el proceso de KDD, con la aplicación de algoritmos específicos para la extracción de conocimiento deseable de conjuntos de datos para un fin en particular (Olson & Delen, 2008). Razón por la cual este capítulo define cada una de las evoluciones de la familia del dataset DARPA y describe someramente cada uno de ellos.

- **DARPA KDD 1998:**

Según (Revathi & Malathi, 2013)el dataset ***DARPA1998***contiene un conjunto de ataques realistas, integrados a un conjunto de conexiones normales, lo cual suministra el insumo de datos que permite evaluar las falsas alarmas y las tasas de detección de IDS. DARPA KDD 1998 consta de dos (2) partes para evaluar la de detección de intrusos: una evaluación Fuera de línea y una en línea, esta última puede ser consultada en (LL-MIT, 2016).

Para la evaluación en tiempo real los sistemas de detección de intrusión fueron entregados al *Air Force Reserch Laboratory–AFRL*. Estos sistemas fueron insertados en el banco de pruebas de red del AFRL para identificar sesiones de ataque en medio de actividades normales, en tiempo real.

- **DARPA KDD 1999:**

Consta de dos partes para la evaluación de detección de intrusos: una evaluación fuera de línea y una en línea los sistemas de detección de intrusos fueron mediante la evaluación fuera de línea, la evaluación en tiempo real, o ambos, (LL-MIT, 2016). Lo cual se le adicionó a la base de datos de tráfico de red ya existente los siguientes tipos de conexiones: tráfico procedente de ordenadores *Windows NT*, ataques procedentes de la red interna, archivos de sistema dump (desde sistema de ficheros que incluyen logs de auditoria de *Windows NT*) y archivos de *Sniffing* (que proporcionan datos rastreados de la red Interna).

En esta ocasión, los datos recolectados producto del proceso de simulación del tráfico real de la red computacional permitieron crear dos (2) subconjuntos: tres semanas de datos de entrenamiento (proporcionados para facilitar el entrenamiento de los sistemas de detección de anomalías) y dos semanas de datos de test (basados en ataques de red,

en medio de actividad normal en background). Una mayor descripción de los subconjuntos anteriormente descritos y los archivos en formato ".tar" que contienen el tráfico de red, pueden ser descargados de (LL-MIT, 2016).

- **DARPA KDD 2000**

Según (LL-MIT, 2016) se obtuvo partiendo de 3 escenarios:

1. **LLDOS:** Este escenario de ataque se lleva a cabo en varias sesiones de red y auditoría. Estas sesiones se han agrupado en 5 fases de ataque, en el transcurso de la cual el atacante explora la red, rompe en un sistema principal mediante la explotación de la vulnerabilidad *Solaris sadmind*, instala el software *DDoS mstream* troyano, y lanza un ataque *DDoS* en un servidor fuera del sitio del anfitrión comprometido.
2. **LLDOS:** Este es el segundo caso de un ataque de datos establecido para crear DARPA. Incluye un ataque de denegación de servicio a cargo de un atacante que es más cauteloso que el atacante en el primer conjunto de datos. El atacante se considera todavía un principiante, ya que el ataque es un guion sobre todo de una manera más cauteloso, es algo que cualquier atacante podría ser capaz de descargar y ejecutar.

    Este escenario de ataque se lleva a cabo en varias sesiones de red y auditoría. Estas sesiones se han agrupado en cinco (5) fases de ataque, en el transcurso de la cual el atacante explora la red, rompe en un sistema principal mediante la explotación de la vulnerabilidad *Solaris sadmind*, instala el software *DDoS mstream* troyano, y lanza un ataque DDoS en un servidor fuera del sitio desde el host comprometido.

3. **CONJUNTO DE ATAQUES NT:** Un experimento con un nivel superior de auditoría NT a la que se ejecuta en la evaluación de 1999 la cual se llevó a cabo en enero de 2000. En este dataset se presentan las tasas de auditorías recogidas del tráfico de un día y el ataque que afecta la *maquina NT*.

- **DARPA NSL- KDD**

Según(Dhanabal & Shantharajah, 2015) el conjunto de datos NSL-KDD es la versión refinada del conjunto de datos del KDD99 debido a las siguientes 4 mejorías:

1. Se eliminan registros redundantes.
2. Se procuró que el número de registros seleccionados de cada grupo de nivel de dificultad sea inversamente proporcional al porcentaje de registros en el conjunto original de datos KDD.
3. Que el número de registros que contiene el dataset sea manejable.
4. Genera razonables tiempos de procesamiento de la información.

2.6 Proceso de selección de características

En los últimos años un gran número de dataset con alto contenido de características se ha hecho públicamente disponible en internet, aumentando su uso en la comunidad científica, ya que para los métodos de aprendizaje automático es demandante analizar un gran número características.

Para resolver este problema, los algoritmos de selección de características se han convertido en un elemento necesario dentro del proceso de aprendizaje, (Sánchez-maroño, 2013) lo define como el proceso de detección de las características relevantes y descartar

las irrelevantes, su objetivo es reducir el tamaño de los datos de entrada para facilitar el procesamiento y análisis de dicha información.

Según (Mendoza Palechor, 2013), la capacidad de emplear la selección de características, es primordial para realizar un análisis eficaz, debido a que los datos contienen información que no es necesaria para la generación del modelo.

Si mantiene las columnas innecesarias durante la generación del modelo, se necesitará mayor capacidad de procesamiento y memoria durante el proceso de entrenamiento, y más espacio de almacenamiento para el modelo completo.

Existen dos enfoques principales en la selección de características(Sánchez-Maroño, 2013):

- **Evaluación individual:** La evaluación individual también se conoce como ranking. Evalúa las características individuales asignándoles pesos de acuerdo a su grado de relevancia.
- **subconjunto individual:** La evaluación subconjunto individual, produce subconjuntos candidatos de características, en base a una cierta estrategia de búsqueda.

Además de esta clasificación, los métodos de selección de características también se pueden dividir en tres modelos según (Spolaˆ & Monard, 2014):

1. **Filtros:**

Se basan en las características generales de los datos de entrenamiento y llevan a cabo el proceso de selección de características como una etapa de preprocesamiento con independencia del algoritmo de inducción.

2. **Envoltorios:**

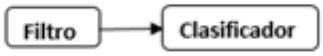
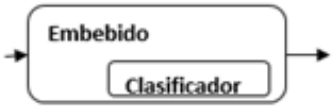
   Consiste en la optimización de un predictor como parte del proceso de selección.

3. **Métodos Embebidos:**

   Realizan la función de selección en el proceso de aprendizaje y suelen ser usados en la implementación de técnicas de aprendizaje de máquina.

Estos tres (3) modelos agrupan diferentes técnicas de selección de características, la Tabla **No 2.4.** Muestra ejemplos de cada uno de ellos.

Tabla No. 2. 4. Técnicas de selección de características según el método

Método	Ventajas	Desventajas	Ejemplos
Filtro 	Independecia del clasificador	No interación con el clasificador	CFS
	Menor costo computacional		Relief
	Rapidos		Md
	Buena capacidad de generalización		Information Gain
			Mrmr
Embebido 	Interacción con el clasificador	La seleccíon depende del clasificador	Fs-Perceptron
	Bajo costo computacional		
	Capta caracteristicas de dependencia		SVM-RFE
Envoltorio 	Iteracción con el clasificador	Riesgos computacionales de sobre ajustes	Wrapper , C4.5
	Dependencia de funciones de capturas	La seleccíon depende del clasificador	Wrapper-SWM

2.6.1. Técnicas de selección de características en sistemas de IDS

La selección de características hace referencia a un concepto utilizado en minería de datos con el objetivo de reducir el tamaño de los datos de entrada, para facilitar el procesamiento y análisis de dicha información. La selección de características no solo tiene en cuenta la disminución de la cardinalidad, es decir, mantener un límite parcial o predefinido en la cantidad de atributos tenidos en cuenta al crear un modelo, además permite descartar de forma adecuada los atributos en función de la utilidad para la realización de un buen proceso de análisis.

Según (R. Kaur, G. Kumar y K. Kumar, 2015) la técnica de selección de características basada en filtrado *(filter*) se utiliza para encontrar el mejor subconjunto características del conjunto original, los métodos de filtrado parecen ser buenos en la selección de un gran subconjunto de datos, no dependen del algoritmo de clasificación y su costo computacional es menor para grandes conjuntos de datos. Las técnicas de selección de características basadas en envoltorios (*wrappers*) definidas en (R. Kaur, G. Kumar y K. Kumar, 2015), utilizan la predicción del rendimiento del algoritmo de aprendizaje para la selección de las características. Mejora los resultados de los predictores correspondientes, y logra mejores tasas de reconocimiento, en algunos casos superando a las técnicas basadas en filtros, sin embargo, dependen del algoritmo de clasificación y para un conjunto de datos grandes, el costo computacional es mayor en el método de *wrapper*.

Por último, los métodos embebidos (*embedded*), también definidos en (R. Kaur, G. Kumar y K. Kumar, 2015), se basan en la evaluación del desempeño de la métrica calculada directamente de los datos, sin referencia directa a los resultados de los sistemas de análisis de datos, en ellos hay una unión de las técnicas de selección de características

con el proceso de aprendizaje para un algoritmo de aprendizaje determinado. Los métodos *embedded* son menos propensos al sobreajuste (*overfitting*) y también dependen del algoritmo de clasificación.

La capacidad de emplear la selección de características es primordial para realizar un análisis eficaz, debido a que los datos contienen información que no es necesaria para la generación del modelo. En esta investigación se han seleccionado las técnicas INFO.GAIN a raíz de que en la exploración preliminar del estado del arte se observó que al implementarlas en temas relacionados a la detección de anomalía sus resultados fueron prometedores más sin embargo no ha sido implementado en técnicas de entrenamiento como lo son *GHSOM, Redes Bayesianas, Naive Bayes, NBTREE, RANDOM FOREST, C4.5 , LOGISTIC y PART* con el objetivo de analizar las métricas de desempeño que arroje el modelo propuesto.

A continuación, se presenta en detalle la técnica de selección de características utilizada en el modelo propuesto en esta investigación

2.6.3. INFO.GAIN

Es una técnica de selección de características basadas en filtros y definida en (Hota & Shrivas, 2014). Es también conocida como ***information gain*** y se utiliza para identificar el nivel de relevancia o ranking de las características de una colección de datos. La ecuación **No. 2.12** define este nivel de relevancia. El atributo con la mayor ganancia de información se elige como el atributo de división para el nodo N.

Este atributo minimiza la información necesaria para clasificar las duplas en la partición resultante y refleja la menor aleatoriedad o impureza en estas particiones.

$$IG(D, X_3) = entropy(D) - \sum_v \frac{|D_v| entropy(D_v)}{|D|} \quad (2.12)$$

Implementando la técnica de selección de características anteriormente documentada al conjunto de datos DARPA KDD-Train contenido al 100%, se genera un resultado de 15 características.

2.7Minería de datos como método para desarrollo de sistema de detección de anomalías

La minería de datos es el proceso de "extracción no trivial de conocimiento implícito, previamente desconocido y potencialmente útil, a partir de los datos (Frawley, Piatetsky-Shapiro, & Matheus, 1992).

Según (Forbes, Evans, Hastings, & Peacock, 2011), la minería de datos es "el descubrimiento automático de patrones o modelos interesantes y no obvios escondidos en una base de datos, los cuales tienen un gran potencial para contribuir en los aspectos principales del negocio". Según (Fayyad, Piatetsky-Shapiro, & Smyth, 1996) la minería de datos "es un mecanismo de explotación de datos consistente en la búsqueda de información valiosa en grandes volúmenes de datos".

Luego de citar las definiciones anteriores y analizar los conceptos, se puede definir la minería de datos como la integración de patrones que deben generar conocimiento, el cual resulta útil para un objetivo en particular basándose en la implementación de métodos de entrenamiento.

2.7.1 Redes neuronales GHSOM (Growing Hiererchical Self Organizing Maps)

Según (Rauber, Pampalk, & Merkl, 2002), GHSOM es una estructura jerárquica y dinámica, desarrollado para superar las debilidades y problemas que presenta SOM. La estructura GHSOM consiste en múltiples capas compuestas de varias SOM independientes cuyo número y tamaño se determinan durante la fase de entrenamiento. El proceso de crecimiento de adaptación está controlado por dos parámetros que determinan la profundidad de la jerarquía y la amplitud de cada mapa. Por lo tanto, estos dos parámetros son los únicos que tienen que ser fijados inicialmente en GHSOM.

Este tipo de mapas nacen como una versión mejorada de la arquitectura SOM, según (Dittenbach, Merkl, & Rauber, 2000) hay dos propósitos para la arquitectura de GHSOM:

1. SOM tiene una arquitectura de red fija, es decir el número de unidades de uso, así como la distribución de las unidades tiene que ser determinada antes del entrenamiento.
2. Los datos de entrada que son de naturaleza jerárquica deberían estar representados en una estructura jerárquica para mayor claridad de la representación. GHSOM utiliza una estructura jerárquica de varias capas, donde cada capa está formada por un número de SOM independientes. Solo un SOM se utiliza en la primera capa de la jerarquía.

Por cada unidad del mapa, una SOM podría añadirse a la siguiente capa de la jerarquía. Este principio se repite con el tercer nivel del mapa y las demás capas de la GHSOM, tal como se muestra en la Figura **No 2.7**.

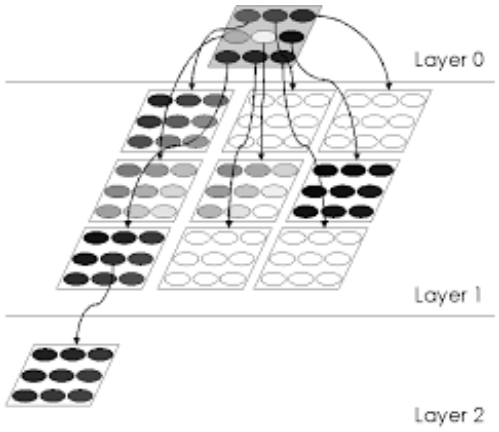


Figura No. 2. 7. Estructura de Una Red GHSOM

Fuente: (A. Rauber, D. Merkl, and M. Dittenbach, 2016)

Según (Dittenbach et al., 2000)por cada unidad del mapa, una SOM podría añadirse a la siguiente capa de la jerarquía. Este principio se repite con el tercer nivel del mapa y las demás capas de la GHSOM para llevar a cabo el proceso de inserción de columnas o filas en un GHSOM deben seguirse los siguientes pasos:

1. Los pesos de cada unidad se inicializan con valores aleatorios.
2. Se aplica el algoritmo estándar de SOM.
3. La unidad con la mayor desviación entre el vector de pesos y los vectores de entrada es elegida para representar la unidad de error.
4. Una fila o una columna se inserta entre la unidad de error y la unidad vecina más distinta en términos de espacio de entrada.
5. Los pasos 1 al 3 se repiten hasta que el Error de Cuantificación Medio (MQE) alcanza un determinado umbral, una fracción del error de cuantificación promedio de la Unidad i, en la capa de procedimiento de la jerarquía. Tal como se aprecia en la Figura **No 2.8**.

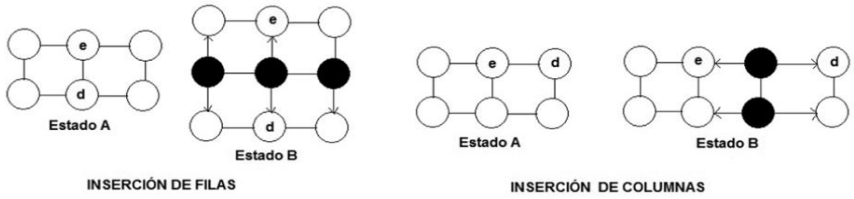


Figura No. 2. 8.Proceso de inserción de filas de una red GHSOM

Fuente: (A. Rauber, D. Merkl, and M. Dittenbach, 2016)

2.7.2 Clasificadores bayesianos

Los clasificadores Bayesianos están basados en la formulación del Teorema de Bayes (2.13) en 1763 (Bayes, Price, & Canton, 1763)

$$P(A|B) = \frac{p(A)p(B|A)}{p(B)} \quad (2.13)$$

donde:

- p(A) es conocido como la probabilidad a priori de que el suceso A sea cierto.
- p(A|B) es conocido como la probabilidad a posteriori, o, la probabilidad de que el suceso A sea cierto tras considerar B. p(B|A) es conocido como verosimilitud o likelihood, e indica la probabilidad de que el suceso B sea cierto, asumiendo que A lo es.
- p(B) es la probabilidad a priori de que el suceso B sea cierto. Actúa de coeficiente normalizador o estandarizados en la fracción.

Este teorema no sólo puede ser aplicado a sucesos, sino también a variables

aleatorias, tanto unidimensionales como multidimensionales. Su formulación general es según formula (2.14):

$$p(Y = y| X = x) = \frac{p(Y=y)p(X=x|Y=y)}{\sum_{i=1}^{r_y} p(Y=y_i)(p(X=x\ |Y=y_i)} \quad (2.14)$$

Aplicado al problema de clasificación supervisada, tenemos que Y = C es una variable unidimensional; mientras que X = (X_1, $X_{2,...}X_n$) es una variable n-dimensional. "X" será la variable predictora, e "Y" la variable a predecir (la clase predicha por el modelo.

Asumiendo una función de error 0/1, un clasificador Bayesiano x) asigna la clase con mayor probabilidad a posteriori dada una determinada instancia, es decir,

$$\gamma(x) = \underset{c}{\text{argmax}}\ p(c \mid x_1, x_{2,\ldots,} x_{n)} \quad (2.15)$$

donde según (2.15), c representa la variable clase, y x_1, $x_{2,...,}x_n$ son los valores de las variables predictoras. Podemos expresar la probabilidad a posteriori de la clase de la siguiente manera:

$$p(c \mid x_1, x_{2,\ldots,} x_n \propto p(c)\ p(x_1, x_{2,\ldots,} x_n \mid c) \quad (2.16)$$

Asumiendo diferentes factorizaciones para p(x_1, $x_{2,...,}$ x_n | c) se puede obtener una

jerarquía de modelos de creciente complejidad dentro de los clasificadores Bayesianos, hasta ordenes exponenciales de 2^{mxn} siendo m y n el número de dimensiones de las dos variables aleatorias.

2.7.3 Naive Bayes

Según (Bayes et al., 1763) es una técnica de clasificación descriptiva y predictiva basada en la teoría de la probabilidad del análisis de T. Bayes. Esta teoría supone un tamaño de la muestra asintóticamente infinito e independencia estadística entre variables independientes, refiriéndose en nuestro caso a los atributos, no a la clase. Con estas condiciones, se puede calcular las distribuciones de probabilidad de cada clase para establecer la relación entre los atributos (variables independientes) y la clase (variable dependiente). Concretamente, dado el ejemplo x = (x_1,..., x_n), donde xi es el valor observado para el i-ésimo atributo, la probabilidad a posteriori de que ocurra la clase y_m teniendo k valores posibles (y_1,..., y_k), viene dada por la regla de Bayes (2.17):

$$P(y_m | x_1 \ldots, x_n) = \frac{P(y_m) \prod_{i=1}^{n} P(x_i | y_i)}{P(x1,\ldots,xn)} \qquad (2.17)$$

donde $P(y_m)$ es la proporción de la clase y_m en el conjunto de datos; e igualmente, P(i) se estima a partir de la proporción de ejemplos con valor x_i cuya clase es y_m. Como podemos deducir, el cálculo de P($_i$ $_i$) obliga a que los valores xi sean discretos, por lo que si existe algún atributo continuo, éste debe ser discretizado previamente. Aplicando (1.1), la

clasificación de un nuevo ejemplo x se lleva a cabo calculando las probabilidades condicionadas de cada clase y escogiendo aquella con mayor probabilidad. Formalmente, si Y = (y1,...,yk) es el conjunto de clases existentes, el ejemplo e será clasificado con aquella clase y_m que satisface la expresión (2.18):

$$\forall\, j \neq i \; P(y_i \,|x_1, \ldots x_n) \; > P(y_j|x_1, \ldots x_n)$$

El clasificador bayesiano es un método sencillo y rápido. Sin embargo, para estimar el término $P(y_m| x_{1...,}x_n)$ es decir, las veces en que para cada categoría aparecen los valores del ejemplo x, se debe recorrer todo el conjunto de entrenamiento. Este cálculo resulta impracticable para un número suficientemente grande de ejemplos por lo que se hace necesario simplificar la expresión. Para ello se recurre a la hipótesis de independencia condicional con el objeto de poder factorizar la probabilidad.

2.7.4 Redes Bayesianas

Una red bayesiana es un grafo acíclico dirigido y anotado que describe la distribución de probabilidad conjunta que gobierna un conjunto de variables aleatorias. Sea $X = \{X_1, X_2,..., X_n\}$ un conjunto de variables aleatorias. Formalmente, una red Bayesiana para X es un par B = <G, T> en el que:

- G es un grafo acíclico dirigido en el que cada nodo representa una de las variables $X_1, X_2,..., X_n$, y cada arco representa relaciones de dependencia directas entre las variables. La dirección de los arcos indica que la variable „apuntada“ por el arco

depende de la variable situada en su origen.

- T es un conjunto de parámetros que cuantifica la red. Contiene las probabilidades $P_B(x_i |)$, para cada posible valor x_i de cada variable X_i y cada posible valor de , donde éste último denota al conjunto de padres de X_i en G. Así, una red bayesiana B define una distribución de probabilidad conjunta única sobre X dada por Friedman, Tepley, Castleberg, & Roe, 1997), como se muestra en (2.19).

$$P_B(X_1, X_2 \ldots X_n) = \prod_{i=1}^{n} P(X_i \mid \Pi_{xi}) \qquad (2.19)$$

Lo que es lo mismo la distribución conjunta de los valores del nodo puede ser escrita como el producto de las distribuciones locales de cada nodo y sus padres. Si el nodo Xi no tiene padres, su distribución local de probabilidad se toma como incondicional, en otro caso se considera condicional.

La topología o estructura de la red no sólo proporciona información sobre las dependencias probabilísticas entre las variables, sino también sobre las independencias condicionales de una variable o conjunto de ellas dada otra u otras variables. Cada variable es independiente de las variables que no son descendientes suyas en el grafo, dado el estado de sus variables padre.

La inclusión de las relaciones de independencia en la propia estructura del grafo hace de las redes bayesianas una buena herramienta para representar conocimiento de

forma compacta – se reduce el número de parámetros necesarios-. Además, proporcionan métodos flexibles de razonamiento basados en la propagación de las probabilidades a lo largo de la red de acuerdo con las leyes de la teoría de la probabilidad.

Para utilizar una red bayesiana como clasificador, un algoritmo de búsqueda determinado encuentra una red B, $P_B(A_1,A_{2,...,}An; C)$, que mejor ajusta a un conjunto de entrenamiento D de acuerdo a alguna función de evaluación (Friedman, Tepley, Castleberg, & Roe, 1997), (Cooper & Herskovits, 1992) . Una vez que se determina la red, B selecciona la etiqueta c que maximiza la probabilidad posterior $P_B(c| a_1,..., a_n)$ (Friedman et al., 1997), (Cooper & Herskovits, 1992) .

2.7.5 Algoritmo C4.5

El algoritmo C4.5 fue desarrollado por Quinlan en 1993 (Quinlan., 1993) como una extensión (mejora) del algoritmo ID3 que desarrolló en 1986. Este algoritmo introduce las siguientes mejoras:

1.Permite trabajar con valores continuos para los atributos, separando los posibles resultados en 2 ramas Ai<= N y Ai> N.

2.Los árboles son menos frondosos, ya que cada hoja cubre una distribución de clases no una clase en particular.

3.Utiliza el método "divide y vencerás" para generar el árbol de decisión inicial a partir de un conjunto de datos de entrenamiento.

4.Se basa en la utilización del criterio de proporción de ganancia (gain ratio), de esta manera se consigue evitar que las variables con mayor número de posibles valores salgan beneficiadas en la selección.

C4.5 produce un árbol de decisión similar al de ID3, con la salvedad de que puede incluir condiciones sobre atributos continuos. Así, los nodos internos pueden contener dos tipos de decisión según el dominio del atributo seleccionado para la partición. Si el atributo Ai es discreto, la representación es similar a la de ID3, presentando una decisión con una condición de salida (rama Ai = Aiv) por cada valor Aiv diferente del atributo.

Si el atributo Ai es continuo, el test presenta dos únicas salidas, Ai ≤ N y Ai> N que comparan el valor de Ai con el umbral N. Para calcular N en la Formula *(2.20)*, se aplica un método similar al usado en (L. Breiman, 1984), el cual ordena el conjunto de t valores distintos del atributo Ai presentes en el conjunto de entrenamiento, obteniendo el conjunto de valores {ai1, ai2,…, ait}. Cada par de valores consecutivos aporta un posible umbral:

$$N = \frac{a_{iv}+ai(v+1)}{2} \qquad (2.20)$$

2.7.6 Algoritmo Naive Bayes Tree

La variante NBTREE (Naive-Bayes Tree) propuesta en (Kohavi, 1996) presenta un algoritmo híbrido entre los árboles de clasificación y el clasificador naive Bayes. Se puede definir NBTREE como un árbol de clasificación cuyas hojas son clasificadores naive-Bayes como muestra la figura (2.21):

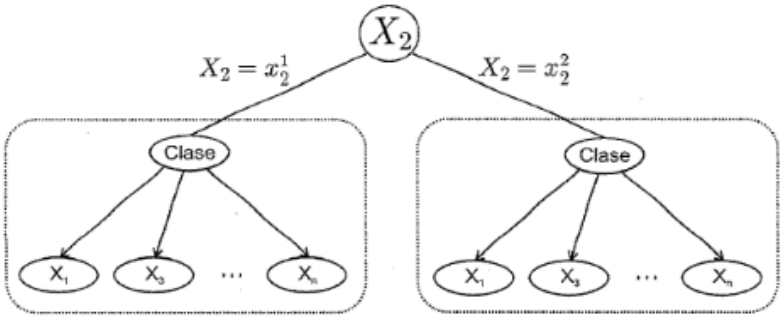


Figura No 2.21: NBTree con un nodo de decisión (X_2) y 2 clasificadores NB como hojas.

Fuente : (Kohavi, 1996)

Cada hoja del NBTREE contiene un clasificador naive-Bayes local que no considera las variables que se encuentran involucradas en la decisión que está en el camino que lleva hasta la hoja. Las propiedades de este árbol de clasificación son: cada nodo interno representa un atributo, cada nodo interno tiene tantos hijos o ramas salientes como valores tiene el atributo representado en dicho nodo, todas las hojas están al mismo nivel y en cualquier camino que se recorra desde la raíz hasta las hojas no existen variables repetidas.

La condición “todas las hojas están al mismo nivel” se impone para simplificar el modelo, pero puede ser no tenida en cuenta en la práctica. Este algoritmo está basado en la verosimilitud marginal de los datos para realizar la búsqueda.

2.7.7 Algoritmo Partial Decision Tree: “Part”

El algoritmo PART de aprendizaje de reglas basado en árboles de decisión parciales (Frank & Witten, 1998) representa un enfoque alternativo híbrido para la inducción de reglas. Básicamente construye una regla, elimina las instancias que ésta cubre y continúa creando reglas recursivamente para las instancias que permanecen hasta que no quede ninguna, pero para crear una regla, se construye un árbol de decisión podado a partir del

conjunto activo de instancias, la hoja de éste con mayor cobertura se convierte en una regla, y se desecha el árbol (recordemos que como se citó en el sub-apartado árboles de decisión, un árbol de decisión se puede ver como un conjunto de reglas si-entonces).

Aunque el hecho de construir repetidamente árboles de decisión para simplemente descartar la mayoría de ellos pueda resultar un tanto extraño, en verdad resulta que el empleo de un árbol podado para obtener una regla en vez de construirla incrementalmente añadiendo conjunciones evita la tendencia a la "sobrepoda". Construir un árbol de decisión completo para obtener una única regla supondría un enorme derroche de recursos, pero en el caso del algoritmo PART: la idea clave es construir un árbol de decisión parcial en vez de uno completo.

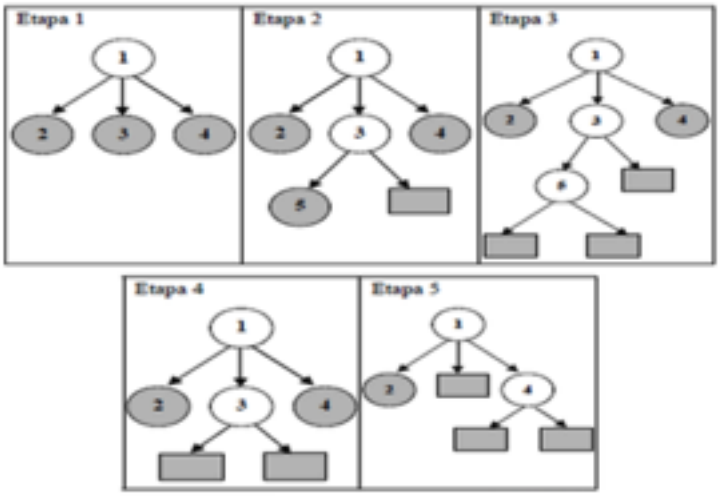
Un árbol de decisión parcial contiene algunas ramas que representan subárboles no definidos. Para generar tal árbol parcial, se integran las operaciones de construcción y poda con el objetivo de encontrar un subárbol "estable" que no pueda simplificarse más. Una vez hallado este subárbol, la construcción del árbol cesa y dicho subárbol se convierte en una regla.

Para la construcción del árbol se procede igual que en el algoritmo de construcción de árboles C4.5, se escoge un atributo-nodo para ser dividido y se evalúa su entropía, los subconjuntos resultantes se expanden en orden creciente de acuerdo con su entropía, empezando con el de menor entropía, debido a que es más probable que la expansión de los subconjuntos de baja entropía finalice rápidamente y dé lugar a subárboles de pequeño

tamaño y por lo tanto a reglas más generales.

La expansión se va realizando recursivamente, pero tan pronto como aparezca un nodo interno cuyos hijos ya se hayan expandido en hojas, se comprueba si dicho nodo interno puede ser sustituido por una única hoja, esto es, se intenta “podar” ese subárbol, y la decisión acerca de esta poda se toma de la misma manera que en C4.5.

Si el reemplazo se lleva a cabo, se vuelve hacia atrás a explorar los nodos hermanos del nodo reemplazado. Sin embargo, si durante la exploración se encuentra un nodo cuyos hijos no sean todos hojas, los subconjuntos restantes ya no se explorarán y, por tanto, los subárboles correspondientes no serán definidos, deteniéndose automáticamente la generación del árbol. La siguiente figura muestra un ejemplo del proceso:



Fuente: (Frank & Witten, 1998)

Desde la etapa 1 hasta la 3, se lleva a cabo la construcción del árbol recursivamente del modo usual, pero escogiendo para la expansión el nodo con la entropía más baja, en este

ejemplo, el nodo 3 entre las etapas 1 y 3. El resto de nodos circulares todavía no son expandidos.

Los nodos rectangulares representan hojas. Entre las etapas 2 y 3, el nodo rectangular tendrá una entropía más baja que su hermano, el nodo 5, pero no puede ser expandido porque ya es una hoja. Entonces se vuelve hacia atrás y el nodo 5 resulta elegido para su expansión. Cuando se alcanza la tercera etapa existe un nodo cuyos hijos son todos hojas, el nodo 5, y esto desencadena el proceso de poda. Se plantea la posibilidad de reemplazar este subárbol, y se acepta tal reemplazo, lo que conduce a la etapa 4. Ahora se considera el nodo 3 para su reemplazo, y de nuevo es aceptado.

El retroceso continúa y ahora resulta que el nodo 4 tiene una entropía más baja que el 2; entonces el nodo 4 se expande en 2 hojas. Se estudia la posibilidad de su reemplazo, y supongamos que el nodo 4 resulta no ser reemplazado. En este punto, el proceso finalizaría, habiéndose obtenido el árbol parcial de 3 hojas de la etapa 5.

Una vez construido un árbol parcial, se extraerá una única regla a partir de él. Cada una de sus hojas se corresponde con una regla posible, y se escogerá la que cubra el mayor número de instancias, puesto que proporcionará la regla más general. Si se está construyendo un árbol parcial y existen instancias con valor desconocido para alguno de los atributos implicados, su tratamiento será similar al empleado en el algoritmo C4.5. Cuando la lista de decisión obtenida vaya a ser utilizada para clasificar una nueva instancia con atributos desconocidos, se generará una distribución de probabilidad sobre las clases correspondientes a las distintas reglas que le puedan ser aplicadas.

La fracción del caso que se asigna a cada una de estas reglas vendrá dada por el porcentaje de casos de entrenamiento que llegando a la regla son cubiertos por ella. Finalmente, la clase más probable de acuerdo con la distribución de probabilidad así obtenida será la que se asigne a la nueva instancia que se está clasificando.

De acuerdo con los experimentos realizados por sus creadores, el algoritmo PART produce con gran rapidez conjuntos de reglas tan o más precisos que otros métodos rápidos de inducción de reglas. Pero su principal ventaja sobre otras técnicas no es el rendimiento sino la simplicidad, y ello se consigue combinando el método de inducción top-down de árboles de decisión con la estrategia separate-and-conquer de aprendizaje de reglas.

2.7.8 AlgoritmoRANDOM FOREST

Random forest, introducido por Breiman en 1999 (Breiman, 1999), utiliza un conjunto (o bosque) formado por muchos árboles de clasificación. Para clasificar un nuevo objeto, cada árbol en el conjunto lo toma como entrada y produce una salida, su clasificación. La decisión del conjunto de árboles se toma como la clase con mayoría de votos en el conjunto (Breiman, 2001) .En Random Forest cada árbol individual se desarrolla de una manera particular:

1. Dado un conjunto de datos de entrenamiento de cardinalidad N, toma N ejemplos aleatoriamente con repetición (un bootstrap). Este será el conjunto de entrenamiento para crear el árbol.

2. Para crear cada nodo del árbol, se utiliza únicamente una pequeña cantidad de las variables del problema. Si cada objeto tiene M variables de entrada, se determina un

número m << M y para cada nodo del árbol se seleccionan m variables aleatoriamente. La variable más relevante de este subconjunto elegido al azar se usa en el nodo. El valor de m se mantiene constante durante la expansión del bosque.

3. Cada árbol es desarrollado hasta la mayor extensión posible. No se realiza poda. [Bre01] muestra que el error del conjunto de los árboles depende de dos factores:

3.1 La correlación entre dos árboles cualesquiera en el bosque. El incremento en la correlación produce un incremento en el error del bosque. La utilización de un subconjunto de variables elegidas al azar y de un boostrap de datos (remuestreo con reposición) tiende a reducir dicha correlación.

Para cada división de un nodo, no se selecciona la mejor variable de entre todas, sino que se selecciona al azar un subconjunto de variables del tamaño especificado y se restringe la selección de la variable a este subconjunto. De esta forma se incluye una mayor variabilidad de árboles y se reduce la dependencia del resultado con las divisiones precedentes.

3.2 La fuerza de cada árbol individual en el bosque. Un árbol con un error bajo es un clasificador fuerte. El incremento de la fuerza de árboles individuales decrementa el error del bosque. La utilización de árboles sin poda va en este sentido. El Random Forest, establece un ranking de la importancia de las variables en la predicción de la variable respuesta. La cuestión de la medida de importancia de las variables es un punto crucial y delicado porque la importancia de una variable está condicionada a su interacción, posiblemente compleja, con otras variables. El Random Forest calcula dos medidas de importancia distintas.

La primera, denominada MDA (Mean DecreaseAccuracy), se basa en la contribución de la variable al error de predicción, es decir, al porcentaje de mal clasificados. El error de clasificación de cada árbol se calcula a partir de la parte de la muestra que ha quedado excluida de la submuestra utilizada en la construcción del árbol, generada por remuestreo.

Para calcular la importancia de cada una de las variables que aparecen en un árbol se permutan aleatoriamente los valores de esa variable, dejando intactos el resto de variables, y se vuelven a clasificar los mismos individuos según el mismo árbol, pero ahora con la variable permutada. La importancia en ese árbol se calcula como el aumento en el error de predicción resultante. Finalmente se calcula la medida MDA, como la media de estos incrementos en todos los árboles en donde interviene la variable.

La segunda medida de importancia, denominada MDG (Mean Decrease Gini), se calcula a partir del índice de Gini. Éste es el criterio que se utiliza para seleccionar la variable en cada partición en la construcción de los árboles y que comporta una disminución de esta medida. La importancia de una variable en un árbol se mide como la suma de los decrementos atribuidos a esa variable y la importancia final, como la media en todos los árboles.

3. Objetivos

Producto de la exploración del estado del arte, se evidencia que en diversos trabajos se han utilizado diferentes técnicas de selección y extracción de características, y de manera complementaria, se utilizan diferentes técnicas de clasificación, algunas de ellas basadas en redes neuronales artificiales.

La técnica de selección de características INFO.GAIN ha generado buenos resultados en la fase de pre-procesamiento de datos, evidencia de ello son los estudios (Muraleedharan, Parmar, & Kumar, 2010), (Oshima, Nakashima, & Nishikido, 2009) y (Pal & Parashar, 2014).Por otra parte, en cuanto al proceso de clasificación, las técnicas GHSOM , RANDOM FOREST,PART, C4.5 , NAIVE BAYES , REDES BAYESIANAS y NBTREE han generado muy buenos resultados luego de evaluar las métricas de desempeño de tal proceso , evidencia de ello son los resultados en (Franco & García, 2013), (Nagaraja & Bose, 2006) ,(Goldman, 2002), (Amor, Benferhat, & Elouedi, 2004), (Lee, Stolfo, & Mok, 1999) .Sin embargo, producto de una revisión exhaustiva del estado del arte, no se ha encontrado evidencia de la integración de la técnica de selección de características INFO.GAIN, con las técnicas de clasificación anteriormente mencionadas , aplicándolas a la detección de intrusos en redes informáticas. Por todo lo anterior, se plantean en esta investigación, los siguientes objetivos.

3.1 Objetivo General

Proponer un modelo funcional de IDS soportado en la técnica de selección de características INFO.GAIN hibridada con las técnicas de clasificación GHSOM, PART,

REDES BAYESIANAS, RANDOM FOREST, C4.5, NAIVE BAYES,NBTREE para optimizar la detección de intrusos en redes informáticas.

3.2 Objetivos Específicos

El anterior objetivo es alcanzado a partir de la consecución de los siguientes objetivos específicos:

1. Generar un estudio comparativo de las técnicas de clasificación GHSOM, RANDOM FOREST, PART, C4.5, NAIVE BAYES, REDES BAYESIANAS y NBTREE.
2. Desarrollar un prototipo de sistema de detección de intrusos que integre las técnicas de selección de características INFO.GAIN y las técnicas de clasificación GHSOM, RANDOM FOREST, PART, C4.5, NAIVE BAYES, REDES BAYESIANAS y NBTREE.
3. Evaluar el sistema propuesto y determinar la eficacia del mismo mediante la aplicación de diferentes métricas de calidad.

3.3 Pregunta de investigación

Basados en lo anterior, la pregunta que direcciona esta investigación es: ¿Cómo hacer más segura una red computacional teniendo en cuenta la técnica de selección INFOGAIN y distintas técnicas de clasificación de datos en sistemas de detección de intrusos?

4. Modelo de IDS basado en técnicas de selección y clasificación

En este capítulo se describe con detalle la propuesta de un IDS basado en anomalías de red, efectuando inicialmente una breve descripción del modelo, plasmando una estructura funcional en donde se describen los scripts usados para el preprocesamiento y normalización del dataset, y a su vez se describen las fases de entrenamiento y clasificación, por último, se evalúan las métricas de calidad de la propuesta.

Siguiendo el procedimiento expuesto en la sección **No.2** y esquematizado en la Figura **No. 2.5**, el modelo que se usará para la detección de intrusos basados en anomalías de red comprende seis fases: (1) selección del conjunto de datos, (2) pre-procesamiento, (3) selección características, (4) entrenamiento, (5) clasificación y (6) cálculo de métricas de desempeño. Para su aplicación se implementaron varios escenarios de simulación, variando las técnicas de entrenamiento y clasificación, para ello se priorizó la escogencia de las características mediante la implementación del método de selección de características INFO.GAIN.

La figura **No. 4.1** presenta el esquema funcional del modelo, cuyos elementos se describen a lo largo del capítulo.

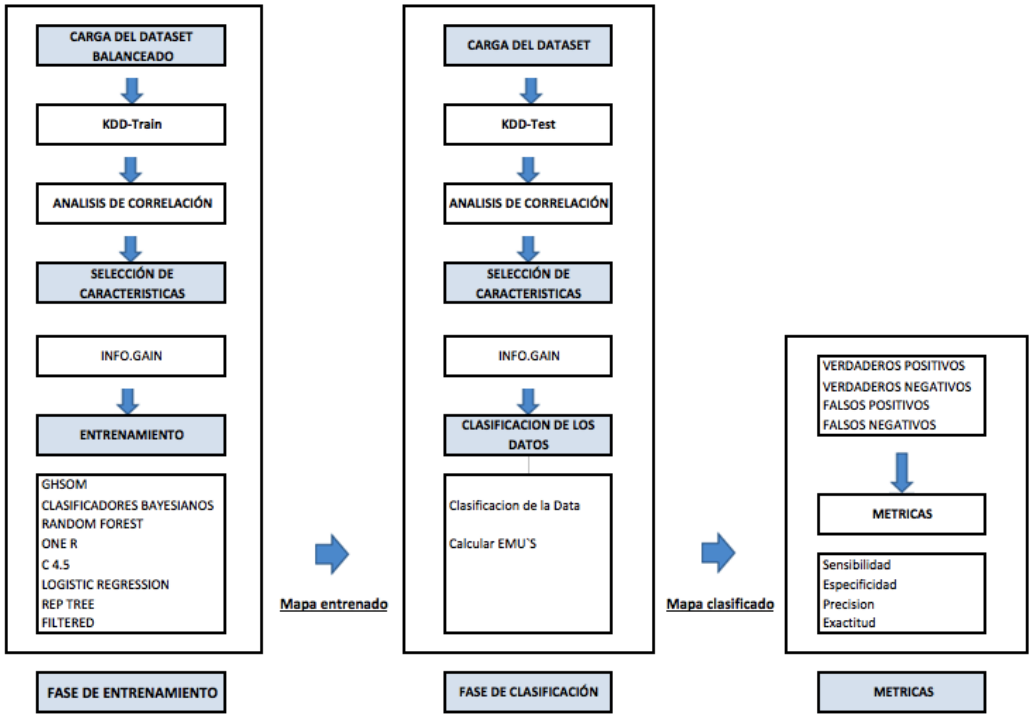


*Figura No. 4. 1.*Modelo funcional propuesto

Fuente: Construcción propia.

4.1 Selección de datos

Muchos investigadores comúnmente se han enfocado en el uso del dataset DARPA para la simulación y análisis de IDS, debido a las ventajas que ofrece en relación a variedad y depuración de sus datos, respecto a otros de su misma familia y otras organizaciones.

Evidencia de su implementación son las múltiples referencias que se hace a este dataset en artículos, paper de conferencias y paper in press, en diferentes bases de datos indexadas. Tal como se evidencia en la Tabla **No 4.1**, donde se muestra la cantidad de artículos por base de datos indexada, en los últimos cinco (5) años que manifiestan haber utilizado el dataset en el ámbito de la detección de intrusos.

Tabla No. 4. 1. Compendio de implementación de uso de DARPA KDD

BASE DE DATO INDEXADA	**2012**	**2013**	**2014**	**2015**	**2016**
SCOPUS	16	16	32	12	16
SPRINGER	10	9	14	11	9
SCIENCEDIRECT	4	3	7	4	6
IEEEXPLORE DIGITAL LIBRARY	11	10	13	7	12

En consecuencia, de lo anterior, en esta tesis, en la etapa inicial de selección de datos, se toma el dataset DARPA, como insumo para la evaluación del modelo propuesto, la cual proviene en formato .txt, con una estructura interna CVS (datos separados por comas). En el preprocesamiento se aplica un método de parsing para gestionar los datos provenientes del dataset original, luego del parseo se procede a la normalización de los datos, pues estos no están homogenizados, y a su vez se aplicó un proceso de balanceo de los datos del dataset (KDD-Train) que incidirá posteriormente en la etapa de aprendizaje.

4.2 Preprocesamiento de datos del dataset

Previa a la fase de selección de características se realizó un preprocesamiento para acondicionar los datos procedentes del dataset DARPA NSL-KDD, específicamente dos acciones, análisis (parsing) y balanceo. Como se denota en el modelo funcional (Figura **No. 4.1**), el preprocesamiento está constituido por dos (2) subfases, las cuales se describen a continuación:

- La fase de parseo se realizó con el objetivo de gestionar los datos procedentes del dataset original, analizando los archivos KDD-Train y KDD-Test, ambos al 100% de sus registros. Este proceso implicó generar una matriz que contiene en cada columna características de cada conexión de red, las cuales son 41 en total y en cada fila las conexiones correspondientes al tráfico de red resultando, 23.486 conexiones para KDD-train y 22.544 conexiones para KDD-Test.
- Después de ejecutar el parsing o parseo se procedió a normalizar los datos del dataset, lo cual implicó la representación de todos los datos contenidos (KDD-Train y KDD-Test) en un formato homogéneo, teniendo en cuenta que dicha colección de datos esta inicialmente representada de forma heterogénea, con atributos en el rango de valores discretos numéricos enteros, continuos numéricos o simbólicos, además algunos atributos tienen un rango de valores muy alto, otros en cambio tienen un rango de valores muy pequeños. Para solucionar esta situación, se desarrolló un algoritmo que representa todos los atributos, lo más homogéneo posible, para que sea posible su uso en procesos posteriores de selección y clasificación, evitando el sobre-entrenamiento de la red neuronal debido a que atributos en un rango de valores mayor podría generar mayor representatividad en el entrenamiento, desestimando atributos que estén representados en valores de rangos menores.

En la propuesta aquí descrita, las 41 características del conjunto de datos NSL-KDD, son utilizadas en el algoritmo SOM y GHSOM según el caso, para el cálculo de las distancias euclidianas, por tanto la escala de estas variables es muy importante para determinar la organización topológica del mapa. Si el rango de valores de una variable es

mucho más grande que el de las otras, ésta probablemente dominará la organización del mapa.

En esta propuesta se utilizó la implementación ***var***, también denominada ***whitening*** o normalización a media cero y varianza unidad. La cual se ha seleccionado debido a que genera un mejor desempeño en las pruebas experimentales respecto a las otras implementaciones. En esta investigación, las variables continuas son normalizadas con media cero y varianza unitaria utilizando la Ecuación **No 2.1**. Por otra parte, todas las variables se escalan en el intervalo [0.1]. Las características simbólicas (fueron codificadas con valores decimales) y las binarias no son normalizadas. Una vez normalizados los datos, se efectuó un análisis comparativo de dos técnicas de selección de características, que se describen en detalle a continuación.

Se efectuó el balanceo de datos por tipo de conexión bi-clase (Trafico normal y Ataque), con lo cual se buscó un equilibrio entre el número de conexiones que hacen referencia al tráfico normal y ataques, dado que si en el entrenamiento un algoritmo de aprendizaje se le suministra muchas más conexiones de un determinado tipo, es posible que durante la fase de clasificación tienda a dar mayor respuesta a un determinado tipo de conexión ya que aprendió la red neuronal más características de esa conexión en la etapa de entrenamiento. Cabe acotar que solo se balanceo el dataset KDD-train, el cual fue utilizado para la fase de entrenamiento, por la razón anteriormente comentada.

4.3 selección de características

En el proceso de selección, de atributos se aplicó la técnica de selección de características INFO.GAIN, con el propósito de realizar un análisis eficaz debido a que los

datos contienen información que no es necesaria para la generación del modelo y así, facilitar el procesamiento y análisis en las siguientes fases.

4.4 Entrenamientode datos del dataset

En la fase de entrenamiento, se construye la red neuronal a partir del análisis de cada una de las muestras del dataset KDD-Train al 100%, por medio de los algoritmos GHSOM, RANDOM FOREST, C4.5, PART, NAIVE BAYES, REDES BAYESIANAS, y NBTREE utilizando el conjunto de datos ya normalizado y reducido a las características de mayor relevancia, para generar un aprendizaje eficiente.

En esta fase, inicialmente se procede a entrenar la red neuronal con las técnicas de aprendizaje anteriormente mencionadas, tomando el conjunto de datos ya normalizado y reducido a múltiples características relevantes para generar así un aprendizaje de alta eficiencia. El proceso de entrenamiento se realiza a partir del conjunto de datos correspondientes al dataset KDD-Train al 100%, producto de ejecuciones iterativas que generan un modelo de aprendizaje.

4.5 clasificación de los datos

Una vez el modelo está entrenado, se procede con la fase de clasificación de los datos, en la cual el clasificador determina que tráfico es normal y cual es un ataque, efectuando la subsiguiente clasificación de cada una de las conexiones del conjunto de datos.

Esta fase se toma como fuente de entrada, la red neuronal ya entrenada con el conjunto de datos KDD-Train al 100% de sus registros. Para efectuar la clasificación cargando el dataset KDD-Test al 100% de sus registros, lo cual representa el proceso de

simulando de ataques reales de red. Esto genera un proceso de clasificación de datos a partir del cual se construyen una nueva colección de datos, generado a partir de los datos de prueba que se obtuvieron mediante el cálculo de las unidades de mejor coincidencia (BMU).

En la fase de clasificación, se cargó el dataset KDD-Test, el cual representa el flujo de datos a clasificar. Este dataset es diferente al conjunto de datos de entrenamiento (KDD-Train). Se reducen las características del KDD-Test y posteriormente se procede a clasificar los datos.

4.6 Evaluación de métricas para validar la calidad de la propuesta

En esta fase final del proceso de simulación, se calculan las métricas de calidad del modelo propuesto, teniendo como argumento de entrada los datos arrojados por la clasificación. Para ello fue necesario calcular la cantidad de: verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN).

A partir de lo anterior se calcularon las métricas (sensibilidad, especificidad, exactitud y precisión) con el propósito de conocer la capacidad de clasificación del modelo propuesto, y permitir validar su eficiencia

5.Escenarios experimentales

En el presente capitulo se recrean y analizan distintos escenarios de simulación para la detección de intrusos en sistemas computacionales aplicando el modelo propuesto utilizando el dataset DARPA NSL-KDD, en el cual se detalla cada escenario aplicando la técnica de selección de características INFO.GAIN y la variación de técnicas entrenamiento y clasificación (GHSOM , RANDOM FOREST,PART, C4.5 , NAIVE BAYES , REDES BAYESIANAS y NBTREE).A su vez se aplica la técnica de ***validación cruzada 10 (diez) pliegues*** como método para sistema de simulación en laboratorio.

En la **sección 5.1** se utilizó validación cruzada con 10 pliegues, de los cuales 9 pliegues para entrenamiento y uno para las pruebas, con el objetivo de efectuar comparaciones entre los diferentes escenarios planteados en el modelo propuesto.

Una vez implementada la técnica de selección, la técnica de entrenamiento y clasificación anteriormente mencionadas se procedió a evaluar unas métricas de desempeño con el fin de identificar la eficiencia del modelo propuesto en el cual se enfocan los mejores resultados en las métricas de desempeño empleadas para valorar la eficiencia del modelo.

En la **sección 5.2** se realiza una consolidación de todos los resultados de simulación con el propósito de identificar el modelo más efectivo con respecto a los mejores resultados de las métricas evaluadas variando la técnica de entrenamiento y clasificación.

5.1 escenarios experimentales con variación de técnicas de entrenamiento y clasificación

Por ello, a continuación, se plantean los mismos escenarios de simulación anteriormente analizados, pero aplicando validación cruzada k-pliegues (k = 10). Ya que k = 10, contiene particiones del 90% de las muestras que fueron seleccionadas aleatoriamente para ajustar el modelo; el resto de las muestras (10%) se utilizó para la prueba.

Estos subconjuntos son diferentes y no comparten ninguna muestra. Este proceso se repitió para los 10 pliegues, asegurándose de que los datos de prueba nunca se hubiesen utilizado en la selección de características o en el entrenamiento del clasificador. Por lo tanto, los resultados proporcionados por los subconjuntos de características seleccionadas y la precisión de la clasificación se calculan como la media de 10.

Si bien es cierto que la exactitud es la proporción de resultados verdaderos (tanto verdaderos positivos, como verdaderos negativos). Esta métrica es, por tanto, la más preponderante para el estudio de tráfico en redes computacionales y es en definitiva de alta relevancia para la toma de decisión de una prueba de laboratorio.

5.1.1 escenarios experimental 1 (conjunto de características seleccionadas, clasificando con ghsom y aplicando validación cruzada)

En el presente escenario de simulación se consideran las 15 características resultado al aplicar la técnica de selección de características INFO.GAIN enel dataset KDD-Train al 100%, y utilizando la técnica de entrenamiento y clasificación de***GHSOM.*** Para la validación cruzada se aplicaron diez (10) pliegues y el resultado de las métricas se calculó a partir de la media de los resultados parciales.

Tabla No. 5.1. *Resultado de prueba de simulación aplicando GHSOM con validación cruzada*

TECNICA	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
GHSOM	96,20%	96,36%	96,52%	96,20%

Como se observa en la tabla No 5.1, plantea unos resultados en sus métricas, con unos índices de precisión de 96,20%, exactitud de 96,36%, sensibilidad en 96,52% , y especificidad con 96,20%.

5.1.2 Escenarios experimental 2 (conjunto de características seleccionadas, clasificando con redes bayesianas y aplicando validación cruzada)

En el presente escenario de simulación se consideran las 15 características resultado al aplicar la técnica de selección de características INFO.GAIN en el dataset KDD-Train al 100%, y utilizando la técnica de entrenamiento y clasificación de ***REDES BAYESIANAS.*** Para la validación cruzada se aplicaron diez (10) pliegues y el resultado de las métricas se calculó a partir de la media de los resultados parciales

Tabla No. 5.2. ***Resultado de prueba de simulación aplicando REDES BAYESIANAS con validación cruzada***

TECNICA	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
BAYES NET	93,12%	94,89%	96,53%	93,35%

Como se observa en la tabla No 5.2, plantea unos resultados en sus métricas, con unos índices de precisión de 93,12%, exactitud de 94,89%, sensibilidad en 96,53%, y especificidad con 93,35%.

5.1.3 Escenarios experimental 3 (conjunto de características seleccionadas, clasificando con naive bayes y aplicando validación cruzada)

En el presente escenario de simulación se consideran las 15 características resultado al aplicar la técnica de selección de características INFO.GAIN en el dataset KDD-Train al 100%, y utilizando la técnica de entrenamiento y clasificación de ***NAIVE BAYES.*** Para la validación cruzada se aplicaron diez (10) pliegues y el resultado de las métricas se calculó a partir de la media de los resultados parciales.

Tabla No. 5.3. ***Resultado de prueba de simulación aplicando NAIVE BAYES con validación cruzada***

TECNICA	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
NAIVE BAYES	84,61%	94,43%	95,76%	86,22%

Como se observa en la tabla No 5.3, plantea unos resultados en sus métricas, con unos índices de precisión de 84,61%, exactitud de 94,43%, sensibilidad en 95,76%, y especificidad con 86,22%.

5.1.4 Escenarios experimental 4 (conjunto de características seleccionadas, clasificando con random forest y aplicando validación cruzada)

En el presente escenario de simulación se consideran las 15 características resultado al aplicar la técnica de selección de características INFO.GAIN en el dataset KDD-Train al 100%, y utilizando la técnica de entrenamiento y clasificación de ***RANDOM FOREST.*** Para la validación cruzada se aplicaron diez (10) pliegues y el resultado de las métricas se calculó a partir de la media de los resultados parciales

tabla no. 5.4. resultado de prueba de simulación aplicando random forest con validación cruzada

TECNICA	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
RANDOM FOREST	99,77%	99,83%	99,89%	99,77%

Como se observa en la tabla No 5.4, plantea unos resultados en sus métricas, con unos índices de precisión de 99,77%, exactitud de 94,83%, sensibilidad en 99,89%, y especificidad con 99,77%.

5.1.5 Escenarios experimental 5 (conjunto de características seleccionadas, clasificando con c4.5 y aplicando validación cruzada)

En el presente escenario de simulación se consideran las 15 características resultado al aplicar la técnica de selección de características INFO.GAIN en el dataset KDD-Train al 100%, y utilizando la técnica de entrenamiento y clasificación de ***C4.5.*** Para la validación cruzada se aplicaron diez (10) pliegues y el resultado de las métricas se calculó a partir de la media de los resultados parciales

Tabla No. 5.5. ***Resultado de prueba de simulación aplicando RANDOM FOREST con validación cruzada***

TECNICA	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
C4.5	99,73%	99,80%	99,86%	99,74%

Como se observa en la tabla No 5.5, plantea unos resultados en sus métricas, con unos índices de precisión de 99,73%, exactitud de 99,80%, sensibilidad en 99,86%, y especificidad con 99,74%.

5.1.6 escenarios experimental 6 (conjunto de características seleccionadas, clasificando con algoritmo part y aplicando validación cruzada)

En el presente escenario de simulación se consideran las 15 características resultado al aplicar la técnica de selección de características INFO.GAIN en el dataset KDD-Train al 100%, y utilizando la técnica de entrenamiento y clasificación con ***ALGORITMO PART.*** Para la validación cruzada se aplicaron diez (10) pliegues y el resultado de las métricas se calculó a partir de la media de los resultados parciales.

Tabla No. 5.6. ***Resultado de prueba de simulación aplicando RANDOM FOREST con validación cruzada***

TECNICA	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
PART	99,75%	99,80%	99,85%	99,75%

Como se observa en la tabla No 5.6, plantea unos resultados en sus métricas, con unos índices de precisión de 99,75%, exactitud de 99,80%, sensibilidad en 99,85%, y especificidad con 99,75%.

5.1.7 Escenarios experimental 7 (conjunto de características seleccionadas, clasificando con nbtree y aplicando validación cruzada)

En el presente escenario de simulación se consideran las 15 características resultado al aplicar la técnica de selección de características INFO.GAIN en el dataset KDD-Train al 100%, y utilizando la técnica de entrenamiento y clasificación con ***NBTREE.*** Para la validación cruzada se aplicaron diez (10) pliegues y el resultado de las métricas se calculó a partir de la media de los resultados parciales.

Tabla No. 5.7. ***Resultado de prueba de simulación aplicando NBTREE con validación cruzada***

TECNICA	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
NB TREE	99,69%	99,79%	99,89%	99,70%

Como se observa en la tabla No 5.7, plantea unos resultados en sus métricas, con unos índices de precisión de 99,67%, exactitud de 99,79%, sensibilidad en 99,89%, y especificidad con 99,70%.

5.2 Consolidado de resultados experimentales

Tabla No. 5.8. ***Resultado de prueba de simulación aplicando NBTREE con validación cruzada***

TECNICA	PRECISION	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
GHSOM	96,20%	96,36%	96,52%	96,20%
BAYES NET	93,12%	94,89%	96,53%	93,35%
NAIVE BAYES	84,61%	94,43%	95,76%	86,22%
RANDOM FOREST	99,77%	99,83%	99,89%	99,77%
C4.5	99,73%	99,80%	99,86%	99,74%
PART	99,75%	99,80%	99,85%	99,75%
NB TREE	99,69%	99,79%	99,89%	99,70%

Como se observa en la tabla 5.8, en donde se comparan los mejores resultados obtenidos de los escenarios de simulación anteriormente expuestos en la sección 5.1. En la cual se llega a deducir que la propuesta basada en la técnica de selección de características ***INFO.GAIN***, y utilizando la técnica de entrenamiento y clasificación ***RANDOM FOREST*** se generan los mejores resultados en las métricas de evaluación a comparación con otros modelos simulados.

Capítulo 6

En el presente capítulo se plasman las conclusiones a las cuales se ha llegado el desarrollo de la tesis de investigación, se anuncian los resultados obtenidos (sección 6.1), se resuelve la pregunta problema (sección 6.2) y a su vez se plantean trabajos futuros (sección 6.3) para así seguir dándole continuidad a la línea de investigación objeto de estudio.

6.1 Conclusiones

Los estudios orientados hacia los sistemas de detección de intrusos son de gran beneficio para garantizar la seguridad de los diferentes tipos de redes informáticas debido a los aportes que generan sus resultados con la aplicación de técnicas de selección de características y clasificación, los cuales son de gran importancia para la construcción de sistemas más eficientes.

Al momento de realizar los distintos escenarios de simulación se llega a determinar que el método de selección de características INFO.GAIN, utilizando la técnica de entrenamiento y clasificación RANDOM FOREST, como se denota en tabla No 5.8, generando asíprecisión de 99,77% , exactitud de 99,83% , sensibilidad de 99,89% y especificidad de 99,77% respectivamente siendo así el modelo más eficiente para la detección de ataques de intrusos basados en anomalías de red.

En relación con lo anterior esta investigación tiene en cuenta métricas muy importantes las cuales generan conocimiento para tener en cuenta en futuras investigaciones.

6.2 Respuesta a la pregunta problema

Para dar respuesta al interrogante planteado: ***¿Cómo hacer más segura una red computacional teniendo en cuenta la técnica de selección INFO? GAIN y distintas técnicas de clasificación de datos en sistemas de detección de intrusos?***

Los estudios orientados a los sistemas de detección de intrusos son de gran beneficio para garantizar la seguridad de los diferentes tipos de redes informáticas, debido a los aportes que generan sus resultados en relación al aumento de las tasas de detección. En particular, los IDS basados en anomalías buscan mejorar estas tasas mediante la aplicación de técnicas de selección de características y clasificación, que permitan la detección sin supervisión.

Dada la indiscutible mejora planteada por el método de selección de característica INFO.GAIN e implementando procesos de aprendizaje y clasificación con (GHSOM, RANDOM FOREST, PART, C4.5 , NAIVE BAYES , REDES BAYESIANAS y NBTREE), utilizando solo 15 características del dataset DARPA NSL KDD-Train al 100% y aplicando validación cruzada con 10 pliegues, los aportes más relevantes de esta investigación han sido:

- La implementación de la técnica INFO.GAIN en procesos de selección de características, que permiten identificar los atributos que más inciden en un posterior proceso de clasificación de tráfico de red en Sistemas de Detección de Intrusos.
- La integración de la técnica INFO.GAIN con la técnica de entrenamiento y clasificación, *RANDOM FOREST*en un modelo funcional que fundamenta futuros desarrollo en materia de sistemas de detección de intrusos.

6.3Trabajos futuros

Teniendo en cuenta los aportes investigativos, metodológicos y prácticos entorno a los sistemas de detección de intrusos basados en anomalías de red aplicando distintas técnicas de entrenamiento y clasificación (GHSOM, RANDOM FOREST, PART, C4.5, NAIVE BAYES, REDES BAYESIANAS y NBTREE), cabe acotar que quedan varios puntos o espacios de trabajo sin explorar que pueden dar a lugar a futuras investigaciones; los cuales son:

Recrear escenarios de evaluación en donde se utilicen distintas técnicas de selección de características (LSA, WRAPPER, GAIN RATIO, RELIEF, COST SENSITIVE, ONE R, FILTERED), implementando las mismas técnicas de entrenamiento y clasificación abordadas en esta investigación con el objetivo de identificar un mejor modelo de detección de intrusos.

Desarrollar un sistema embebido basado en software y hardware libre en donde se implementen los scripts propuestos en esta investigación con el propósito de analizar la efectividad y calidad de los resultados bajo ataques reales en un sistema de red de computadoras.

Analizar alternativas para implementar eficientemente los sistemas de detección de intrusos mediante el uso de módulos optimizados en ordenadores con varios procesadores o tarjetas de interfaz de red programable (Por ejemplo: Dispositivos FPGA).

Referencias

A. *Rauber*, D. Merkl, and M. Dittenbach. Department of Software Technology. Vienna University of Technology. The GHSOM Architecture and Training Process. Consulted: July-2016. Available on-line: http://www.ifs.tuwien.ac.at/~andi/ghsom/description.html

Agosta, G., Barenghi, A., Parata, A., & Pelosi, G. (2012, April). Automated security analysis of dynamic web applications through symbolic code execution. In Information Technology: New Generations (ITNG), 2012 Ninth International Conference on (pp. 189-194). IEEE.

Andersen, J. E., Glasdam, S. M., Larsen, D. B., & Molenaar, N. (2016). New Concepts of Quality Assurance in Analytical Chemistry: Will They Influence the Way We Conduct Science in General?. Chemical Engineering Communications, (just-accepted).

Bace, R., & Mell, P. (2001). Intrusion Detection Systems, 1–51.

Bayes, T., Price, R., & Canton, J. (1763). *An essay towards solving a problem in the doctrine of chances* (pp. 370-418). C. Davis, Printer to the Royal Society of London.

Bell, D. E., & LaPadula, L. J. (1973). Secure computer systems: Mathematical foundations (No. MTR-2547-VOL-1). MITRE CORP BEDFORD MA.

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. Knowledge and information systems, 34(3), 483-519.

Breiman, L. (1999). Random forests. *UC Berkeley TR567*.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Brugger, S. T., & Chow, J. (2007). An assessment of the DARPA IDS Evaluation Dataset using Snort. UCDAVIS department of Computer Science,1(2007), 22.

Cooper, G. F. and E. Herskovits (1992). A Bayesian method for the induction of probabilistic networks from data. Machine Learning 9, 309–347.

Chou, T. S., & Yen, K. K. (2007, June). Fuzzy belief k-nearest neighbors anomaly detection of user to root and remote to local attacks. In *Information Assurance and Security Workshop, 2007. IAW'07. IEEE SMC* (pp. 207-213). IEEE.

De la Hoz Franco, E., De la Hoz Correa, E. M., Ortiz, A., & Ortega, J. (2012). Modelo de detección de intrusiones en sistemas de red, realizando selección de características con FDR y entrenamiento y clasificación con SOM. INGE CUC, 8(1), 85-116.

Dhanabal, L., & Shantharajah, S. P. (2015). A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms, 4(6), 446–452. http://doi.org/10.17148/IJARCCE.2015.4696

Dittenbach, M., Merkl, D., & Rauber, A. (2000, July). The Growing Hierarchical Self-Organizing Map. In *IJCNN (6)* (pp. 15-19).

Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, J., & Tan, P. (2002). Data Mining for Network Intrusion Detection, 21–30.

Eid, H. F., Hassanien, A. E., Kim, T. H., & Banerjee, S. (2013). Linear correlation-based feature selection for network intrusion detection model. InAdvances in Security of Information and Communication Networks (pp. 240-248). Springer Berlin Heidelberg.

Elsayed, T., Asadi, N., Wang, L., Lin, J. J., & Metzler, D. (2010). UMD and USC/ISI: TREC 2010 Web Track Experiments with Ivory. In TREC.

ENGEN, V. (2010). Machine learning for network based intrusion detection (Doctoral dissertation, Bournemouth University).

Ert, L., Eilertson, E., Tan, P., Kumar, V., & Srivastava, J. (n.d.). Chapter 3 MINDS - Minnesota Intrusion Detection System, 1–21.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. AI magazine, 17(3), 37.

Forbes, C., Evans, M., Hastings, N., & Peacock, B. (2011). Statistical distributions. John Wiley & Sons.

Frank, E., & Witten, I. H. (1998). Generating accurate rule sets without global optimization.

Frawley, W. J., Piatetsky-Shapiro, G., & Matheus, C. J. (1992). Knowledge discovery in databases: An overview. AI magazine, 13(3), 57.

Friedman, J.S., C.A. Tepley, P.A. Castleberg, and H.Roe, Middle-atmospheric Doppler lidar using an iodine-vapor edge filter, Optics Letters, 22, 1,648-1,650, 1997.

Guo, C., Zhou, Y., Ping, Y., Luo, S., & Lai, Y. (2013). Efficient intrusion detection using representative instances. Computers & Security, 39, 255–267.

Howard, J. D. (1997). An Analysis of Security Incident on the Internet [Ph D dissertation]..

Hota, H. S., & Shrivas, A. K. (2014). Advanced Computing, Networking and Informatics- Volume 1, 27. http://doi.org/10.1007/978-3-319-07353-8

Hyun, Y., Huffaker, B., Andersen, D., Aben, E., Shannon, C., Luckie, M., & Claffy, K. (2011). The CAIDA IPv4 routed/24 topology dataset. URL http://www. caida.org/data/active/ipv4_routed_ 24_topology_ dataset. xml.

J. R. Quinlan. C4.5: Programs for machine learning. Morgan Kaufmann, San Mateo, California, 1993.

Kendall, K. (1999). *A database of computer attacks for the evaluation of intrusion detection systems*. MASSACHUSETTS INST OF TECH CAMBRIDGE DEPT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE.

Kim, J., & Bentley, P. (1999, July). Negative selection and niching by an artificial immune system for network intrusion detection. In *Proc. of GECCO'99*(pp. 149-158).

Lee, W., Stolfo, S. J., & Mok, K. W. (1999, August). Mining in a data-flow environment: Experience in network intrusion detection. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 114-124). ACM.

Liao, H., Lin, C. R., Lin, Y., & Tung, K. (2013). Journal of Network and Computer Applications Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16–24.

Mendoza Palechor, F. (2013). Aplicación de selección de características, métricas de aprendizaje y reducción de dimensión en sistemas de detección de intrusos, (C).

Muraleedharan, N., Parmar, A., & Kumar, M. (2010). A flow based anomaly detection system using chi-square

Olson, D., & Delen, D. (2008). Advanced data mining techniques.

Oshima, S., Nakashima, T., & Nishikido, Y. (2009). Extraction of Characteristics of Anomaly Accessed IP Packets Using Chi-square Method. 2009 International Conference on Complex, Intelligent and Software Intensive Systems, 287–292. http://doi.org/10.1109/CISIS.2009.116

Panda, M., Abraham, A., & Patra, M. R. (2010, August). Discriminative multinomial naive bayes for network intrusion detection. In Information Assurance and Security (IAS), 2010 Sixth International Conference on (pp. 5-10). IEEE.

Patra, J. C., Abraham, J., Meher, P. K., & Chakraborty, G. (2010, July). An improved SOM-based visualization technique for DNA microarray data analysis. In Neural Networks (IJCNN), The 2010 International Joint Conference on (pp. 1-7). IEEE.

Piattini, M., & Del Peso, E. (2001). Auditoria informática. Un enfoque práctico.

R. Goldman. A Stochastic Model for Intrusions. In Symposium on Recent Advances in Intrusion Detection (RAID), 2002.

R. Kaur, G. Kumar y K. Kumar, «A Comparative Study of Feature Selection Techniques for Intrusion Detection,» de 2nd International Conference on Computing for Sustainable Global Development, 2015.

R. Kohavi. Scaling up the accuracy of naive-Bayes classifiers: a decision-tree

Revathi, S., & Malathi, A. (2013, December). A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection. InInternational Journal of Engineering Research and Technology (Vol. 2, No. 12 (December-2013)). ESRSA Publications.

Robling Denning, D. E. (1982). Cryptography and data security. Addison-Wesley Longman Publishing Co., Inc..

Sabhnani, M., & Serpen, G. (2003, June). KDD Feature Set Complaint Heuristic Rules for R2L Attack Detection. In Security and Management (pp. 310-316).

Sánchez-maroño, V. B. N. (2013). A review of feature selection methods on synthetic data, 483–519. http://doi.org/10.1007/s10115-012-0487-8

Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., & Chaintreau, A. (2006). CRAWDAD dataset cambridge/haggle (v. 2006-09-15). CRAWDAD wireless network data archive.

Sévigny, P., DiFilippo, D., Laneve, T., Chan, B., Fournier, J., Roy, S., ... & Maheux, J. (2010, April). Concept of operation and preliminary experimental results of the DRDC through-wall SAR system. In Proc. SPIE (Vol. 7669, p. 766907).

SINGH, G., ANTONY, D. A., & LEAVLINE, E. J. (2013). DATA MINING IN NETWORK SECURITY-TECHNIQUES & TOOLS: A RESEARCH PERSPECTIVE. Journal of Theoretical & Applied Information Technology,57(2).

Singh, R., Kumar, H., & Singla, R. K. (2013, December). Analysis of Feature Selection Techniques for Network Traffic Dataset. In Machine Intelligence and Research Advancement (ICMIRA), 2013 International Conference on (pp. 42-46). IEEE.

Singh, R., Singh, D., & BUIT, B. (2014). A review of network intrusion detection system based on KDD dataset. IJETS International Journal of Engineering and TechnoScience, 5(1), 10-15.

Spolaˆ, N., & Monard, M. C. (2014). Label Construction for Multi-label Feature Selection. http://doi.org/10.1109/BRACIS.2014.52

Stavroulakis, & Stamp. (2010). " Handbook of Information and Communication Security " What the book is about and like, 1–13.

Vicente Cestero, E., Jiménez Martín, A., & Mateos Caballero, A. (2013). A fuzzy extension of MAGERIT methodology for risk analysis in information systems.

Wu, S., & Banzhaf, W. (2010). The use of computational intelligence in intrusion detection systems: A review. Applied Soft Computing.

Zargar, G. R., & Kabiri, P. (2010). Selection of effective network parameters in attacks for intrusion detection. In Advances in Data Mining. Applications and Theoretical Aspects (pp. 643-652). Springer Berlin Heidelberg.

Zargari, S., & Voorhis, D. (2012). Feature Selection in the Corrected KDD-dataset, Emerging Intelligent Data and Web Technologies (EIDWT). 2012 Third International Conference on.